中学生文库



图论中的几个极值问题

管梅谷

上海教育出版社

内容提要

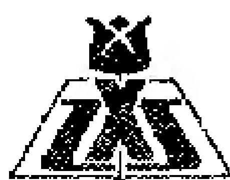
图论是近年来发展较快，用途很广的一个数学分支。特别是图论中的各种极值问题，对于工农业生产更有直接的应用。本书选取了一些基本的图论极值问题，深入浅出地叙述了它的解法。目的是使中学生能够了解这一领域的一些分析问题的思想方法，并学到一些具体解决问题的技能。

中学生文库　图论中的几个极值问题

管梅谷

上海教育出版社出版
（上海永福路123号）
上海市印刷四厂印刷　新华书店上海发行所发行
开本787×1092 1/32　印张5.75　字数131,000
1981年9月第1版　1981年9月第1次印刷
印数1-49,000本

统一书号：7150·2594　定价：0.43元



目录

ZHONG XUE SHENG WENKU

引　　言

图论是近几十年来发展得最快的数学分支之一．为什么会发展得这样快呢？主要是由于在许多科学技术领域中发现了图论的用处，例如物理、化学、计算机科学、运筹学等等．本书不想涉及很多方面，主要想介绍一下图论中的一些极值问题，这些问题用处很广，也比较有趣．

为了让大家知道这本书主要讲些什么，我们先举几个例子．

［例 1］　图 0.1 中画的是一个“图”，当然，究竟什么叫做“图”，以后还要仔细讲．这里只要先记住，我们研究的图，指的是只由点和线组成的图形．我们先把图 0.1 看成是一个公路网，$v_1, v_2, v_3, \cdots, v_{10}$ 是一些城镇，每条线旁边的数字代表这一段公路的长度．现在问，要从 v_1 把货物运到 v_{10} 去，走哪一条路最近？

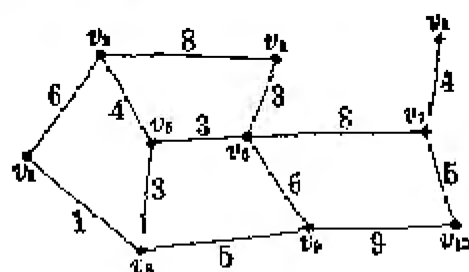


图　0.1

这个问题通常叫做最短路问题．读者不难看出，这是一个有很大现实意义的问题．它不仅出现在各种运输问题中，而且在计算机线路设计等问题中也有用．因为这种问题研究的是“从 v_1 到 v_{10} 的所有路中，哪一条路最短？”因此，它是一

个极大极小问题，即极值问题，而它又和一个"图"密切联系着，因此这种问题就叫做图论中的极值问题。

[例 2]　仍旧看图 0.1，假设有一个巡回售货队要从 v_1 出发，到 v_2，v_3，…，v_{10} 各地去巡回售货，最后再回到 v_1，问他们应该选择怎样的一条路线，才能既走遍所有的城镇，而走的路程又最少？

这类问题国外称为"旅行售货员问题"。它是一个相当困难的问题，用处也很大，邮局里这类问题就很多。例如，一个开信箱取信的同志，他负责开 30 个信箱，就会遇到这类问题。又如专门负责从一个城市的邮政总局向各个分局运送邮件的汽车，也经常遇到这种问题。因此，也有人把这种问题叫做"邮路问题"。

[例 3]　还是把图 0.1 看成公路网，v_1，v_2，…，v_{10} 看成一个公社的城镇。现在要在这些城镇中选择一处来设立卫生院，问设在哪里最"好"？

当然，首先应该明确一下，怎样才算"好"？举一个简单的例子来说，如果有五个城镇，卫生院设在城镇甲处，其他四个城镇的人来看病时，分别要走 7 里、3 里、2 里、1 里。卫生院要是改设在乙处，则另外四个城镇的病人要走 5 里、4 里、4 里、2 里。在这种情况下，设在甲处好还是设在乙处好？一般说来，还是乙处好，因为设在甲时，最远的病人要走 7 里，而设在乙时，最远的只走 5 里。因此，要判定一个城镇设卫生院好不好，可以按来看病的病人中，最远的病人走的距离为标准（也有其他标准，后面还要讨论）。我们的目的是找这么一个城镇，使得最远的病人所走的路尽量少。

例 2、3 中研究的都是与一个图有关的极大极小问题，因

此也都是图论中的极值问题.

上面的三个问题，都是很明显地和一个图联系着的．让我们再来讲一个例子，表面上看，它与图并没有什么关系，但是经过分析，仍然可以归结为图论中的极值问题.

[例 4]　英国某空军部队有若干个驾驶员，专门驾驶一种型号的飞机，这种飞机每架需要两个驾驶员．由于种种原因，例如，语言不通或训练上的问题，有些驾驶员不能在同一架飞机上飞行，问应如何搭配驾驶员，才能使出航的飞机最多.

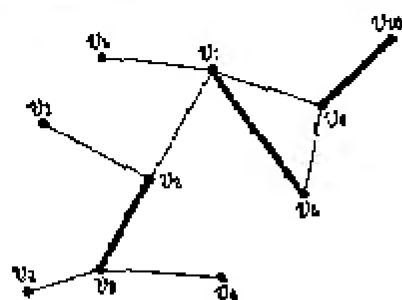


图　0.2

为简单起见，假设有 10 个驾驶员，图 0.2 中的 v_1, v_2, $\cdots$, v_{10} 就代表这 10 个驾驶员．如果两个人可以同机飞行，就在代表他们的两个点之间连一条线；两个人不能同机飞行就不连．从图 0.2 可以看出，v_1 和 v_2 可以同机飞行，而 v_1 和 v_3 就不行．画了这个图后，就可以在图上来研究搭配飞行员的问题了．例如，我们可以组成 $\{v_2, v_8\}$, $\{v_7, v_5\}$, $\{v_6, v_{10}\}$ 这么三组，图 0.2 中画的三条粗线就代表这种搭配方案．注意，任何两条粗线不能有公共的端点，例如，在图 0.2 中，不能再把 v_1 与 v_2 间的连线加粗，因为这样一来，v_2 就既要与 v_1 一起飞，又要与 v_8 一起飞，这当然是不行的．因此，要使出航的飞机最多，就相当于在图 0.2 中把尽量多的线加粗，而加粗的线要两两没有公共端点．今后，我们把一个图中两两没有公共端点的一组线叫做一个“匹配”．这样定义以后，上面的问题就成为：如何找一个包含最多线的匹配？这个问题叫做最大匹配问题．请大家试试看，能不能从图 0.2 中找出一个包含

四条线的匹配．再试试能不能找到包含五条线的匹配．

象例 4 那样经过一些分析后才转化为图论中的极值问题的例子很多，后面还要讲，这里就不多举例了．

讲了上面几个例子，现在可以把这本书的目的说一说了．

这本书主要就是讲这类图论中的极值问题．讲它们是怎样从生产实际中提炼出来的；怎样求解这些问题；哪些问题已经有很好的解法了，哪些还没有等等．当然图论中的极值问题很多，我们只能选一些比较基本的来讲讲．在讲这些问题之前，还要先在第一、二两章中讲一些图论的基本知识．这些知识讲起来可能会有些枯燥，不过大家还是要耐心地看下去．因为这些内容是整个这本书的基础，把基础打好了，后面的内容学起来也就容易些了．

一、无 向 图

1. 集合论的一些基本知识

集合论现在已经几乎是所有数学的基础了. 图论当然也不例外. 因此, 这里先把今后要用到的集合论中一些最基本的概念和符号讲一下.

我们把任意一些对象所组成的总体叫做一个集合. 以后常常用大写英文字母表示集合. 很容易举出一些集合的例子来.

[例 1] 全体正整数的集合 $\boldsymbol{N}$.

[例 2] 全体实数的集合 $\boldsymbol{R}$.

[例 3] 1, 3, 5, 6, 9 五个数的集合 M.

[例 4] 三角形 ABC 的边的集合 L.

[例 5] 方程 $x^2+1=0$ 的所有实数根的集合 P.

[例 6] 教室中所有桌子的集合 Q.

类似的例子还可以写出很多来.

组成一个集合的对象, 叫做这个集合的元素, 例如数 1, 5, 10 都是正整数集合 $\boldsymbol{N}$ 的元素; $\sqrt{2}$, π, -3 是实数集合 $\boldsymbol{R}$ 的元素. 以后常常用小写英文字母表示某一个集合的元素. 如果 a 是集合 A 的元素, 我们就记作:

$$a \in A,$$

读成“a 属于 A”．如果 a 不是 A 的元素，则记作：

$$a \overline{\in} A \text{ 或 } a \notin A,$$

读成“a 不属于 A”．例如：$1 \in \boldsymbol{N}$，$\pi \in \boldsymbol{R}$，但 $\pi \notin \boldsymbol{N}$．

一个集合可以包含无穷个元素，这种集合叫做无限集合，例如前面的例 1、例 2 中的 N 和 R 都是无限集合．只包含有限个元素的集合叫做有限集合，例 3 与例 4 中的集合 M 和 L 都是有限集合．今后我们要用到的主要是有限集合．

再来看看例 5，仔细一看就知道，其实这个集合里一个元素也没有，这种什么也不包含的集合叫做空集合，今后用一个固定的符号 ϕ 来表示空集合．

为了说明一个集合是由哪些元素组成的，可以用以下两种办法．第一种是列举出这个集合的元素，这是在有限集的情况下常用的方法．如例 3 中的 M 可以写成：

$$M = \{1, 3, 5, 6, 9\}.$$

又例如集合 V 由 v_1，v_2，…，v_{100} 这 100 个元素组成，则可以记成：

$$V = \{v_1, v_2, \cdots, v_{100}\}.$$

另外一种办法是说明这个集合中的元素具有什么特征．例如可以把正整数集合 $\boldsymbol{N}$ 记成：

$$N = \{n \mid n \text{ 是正整数}\}.$$

这种写法的意思是：$\boldsymbol{N}$ 是由一些元素 n 组成的，是哪些 n 呢？就是满足括号中竖线后面所讲的性质的那些 n，也就是说，n 必须是正整数．又如例 5 可以记成：

$$P = \{a \mid a \text{ 是 } x^2+1=0 \text{ 的实数根}\}.$$

这两种表示集合的办法各有长处，今后都将用到．

设 A 和 B 是两个集合，如果 A 与 B 包含的元素完全一样，就说 A 与 B 相等，记作：

$$A=B.$$

如果 A 的元素也都是 B 的元素，我们就说 A 是 B 的一个子集合，记作：

$$A\subseteq B.$$

例如，自然数集合 $\boldsymbol{N}$ 是实数集合 $\boldsymbol{R}$ 的子集合，即 $\boldsymbol{N}\subseteq\boldsymbol{R}$. 而例 3 中 M 又是 $\boldsymbol{N}$ 的子集合.

我们还要引入一个概念，叫做真子集合. 就是：如果 $A\subseteq B$，但 A 与 B 又不相等. 就称 A 为 B 的真子集合. 记成：

$$A\subset B$$

这时，至少有一个元素属于 B 但不属于 A.

设 A 是一个集合，今后常常要用到所有以 A 的一对元素 (a_1, a_2) 为元素的集合，我们把这个集合记作 $A\times A$，它的元素叫做 A 的有序元素对. 例如，$A=\{1, 2, 3\}$，则 $A\times A$ 由 $(1, 1)$，$(1, 2)$，$(1, 3)$，$(2, 1)$，$(2, 2)$，$(2, 3)$，$(3, 1)$，$(3, 2)$，$(3, 3)$ 9 个元素组成. 这里 $(1, 2)$ 和 $(2, 1)$ 看成是不同的，这就是把它们叫做"有序元素对"的原因. 另外，我们还要考虑 A 的无序元素对，它和有序元素对的差别在于把 $(1, 2)$ 和 $(2, 1)$ 这样的对象看成是相同的. 因此，当 $A=\{1, 2, 3\}$ 时，A 的无序元素对就只有 6 个. 就是 $[1, 1]$，$[1, 2]$，$[1, 3]$，$[2, 2]$，$[2, 3]$，$[3, 3]$ 了（这里，为了与有序元素对区别，我们用方括号表示无序元素对）. 我们把 A 的所有无序元素对组成的集合记作 $P_2(A)$. A 的有序元素对和无序元素对这两个概念在图论中是很重要的，这一点在下面两节中马上就会看到.

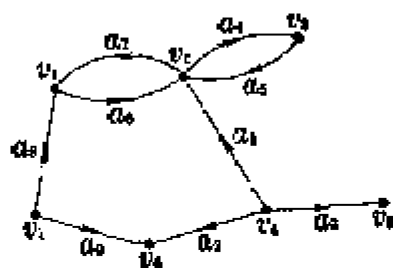
习　题

1. 设$A=\{1, 2, 3\}$，把A的所有子集合都写出来.

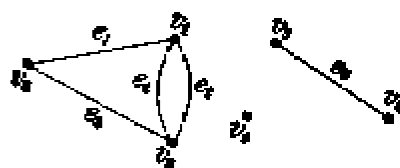
2. 设$A=\{1, 2, 3, 4\}$，写出A的所有有序元素对以及无序元素对.

2. 无向图的定义

我们所讲的图(有些书上也叫线性图)，指的是仅由点和线组成的图形. 例如图1.1中画的就是两个图的实例，这两个图之间还有差别，图1.1(*a*)中的图D，每条线上都有一个箭头以代表这条线的方向，这种图叫有向图，图1.1(*b*)中的图G则叫做无向图.



(*a*) 图D



(*b*) 图G

图 1.1

这一章先讲无向图. 分析一下图1.1(*b*)中的G，可以看出，它包含着6个点：$v_1, v_2, \cdots, v_6$；5条线：$e_1, e_2, \cdots, e_5$，或者说G是由两个集合组成的，一个是

$$V=\{v_1, v_2, v_3, v_4, v_5, v_6\},$$

另一个是

$$E=\{e_1, e_2, e_3, e_4, e_5\}.$$

另外，还可以看出点和线之间有一定的关系，就是每一条线都以两个不同的点为它的端点，例如e_1以v_1和v_2为端点，e_2和

e_4 都以 v_1 和 v_3 为端点等等. 下面的定义 1.1 就是从以上的分析中概括出来的.

定义 1.1 设 $V=\{v_1, v_2, \cdots, v_n\}$ 是空间中 n 个点的集合(这些点可以都在一个平面上, 也可以不在一个平面上), $E=\{e_1, e_2, \cdots, e_m\}$ 是 m 条线的集合, 如果满足:

(1) V 不是空集合;

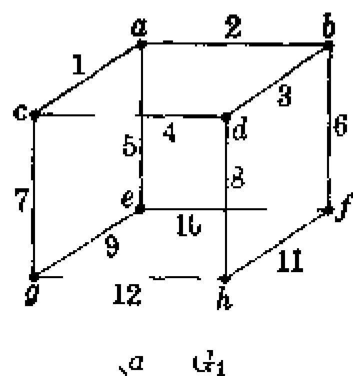
(2) E 的每一条线 e_i 以 V 中的两个不同的点 v_s 与 v_t 为端点.

(3) E 的任意两条线, 除了端点以外, 没有其他的公共点.

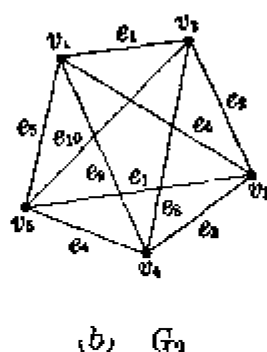
则把由 V 和 E 组成的图形 G 叫做一个无向图, 记作 $G=[V, E]$. 称 V 中的点为 G 的顶点, E 中的线为 G 的边.

因为这一章只讲无向图, 因此, 有时就简单地把一个无向图叫做图.

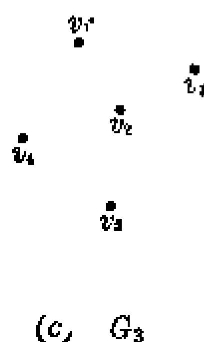
由定义 1.1 可以看出, 图 1.1(b) 中的 G 是一个无向图. 图 1.2 是另外三个无向图的例子. (a)中的 G_1 是由一个立方体的顶点和棱组成的, 它的顶点集合是$\{a, b, \cdots, h\}$, 边集合是$\{1, 2, \cdots, 12\}$. 为简单起见, 今后也常常用一些小写英文字母和数字来代表顶点和边. (b)中的 G_2 的顶点集合是 $\{v_1,$



(a) G_1



(b) G_2



(c) G_3

图 1.2

$v_2, \cdots, v_5\}$，边集合是 $\{e_1, e_2, \cdots, e_{10}\}$. 注意，这里我们把所有对角线看成是互相不相交的，即有些线是从别的线上面飞越过去的；(c) 中的 G_3 只有 5 个顶点，没有边，也可以说，边集合是空集合. 请注意定义 1.1 中要求顶点集合 V 不能是空集合，但边集合可以允许是空集合，因此 G_3 也是一个无向图。

从定义 1.1 可以看出，要确定一个无向图，必须明确指出以下三点.

(1) 它包含哪几个顶点（即集合 V 是由哪些元素组成的）.

(2) 它包含哪几条边（即集合 E 是由哪些元素组成的）.

(3) 每条边以哪两个顶点为端点.

至于这些点的位置如何，连接它们的线是直的还是弯的，都无关紧要. 今后，称上述三点为一个无向图的三个要素.

例如，图 1.2(b) 中的图 G_2 的三个要素是：

(1) $V=\{v_1, v_2, \cdots, v_5\}$,

(2) $E=\{e_1, e_2, \cdots, e_{10}\}$,

(3) 每条边的端点如表 1.1 所示.

对于一个无向图来说，只要这三个要素都知道了，我们就可以认为这个图被完全确定了，并且一定可以把它画出来. 例

表 1.1

边	端 点	边	端 点
e_1	v_1, v_2	e_6	v_1, v_3
e_2	v_2, v_3	e_7	v_3, v_5
e_3	v_3, v_4	e_8	v_2, v_4
e_4	v_4, v_5	e_9	v_1, v_4
e_5	v_5, v_1	e_{10}	v_2, v_5

如，已经知道了一个图 $G=[V, E]$ 中的 $V=\{v_1, \cdots, v_5\}$，$E=\{e_1, \cdots, e_{10}\}$，又知道各条边的端点如表 1.1 中所列的那样．要想把这个图画出来，只要先任意画 5 个点 $v_1, v_2, \cdots, v_5$，再根据表 1.1 的要求，在 v_1 与 v_2 间连一条边 e_1，在 v_2 与 v_3 间连一条边 e_2，…，最后在 v_3 与 v_5 间连一条边 e_{10}．当然，如果在纸上画，这 10 条边中的任意两条除了在端点处可能相遇外，在其他地方也可能相交，但可假定这张图是画在空间的，并且不在交叉点上画黑点，以表示在这点是飞越过去的，这样就把这个图画出来了．当然，这样画出来的图和图 1.2(*b*) 中的图可能很不一样，例如，很可能画成图 1.3 那个样子（请大家试试看也画一个）．但是从本质来看，这两个图是一样的．

在根据一个无向图的三要素画这个图时，我们常常希望能把这个图的所有顶点和边都画在一个平面上，并且使得任何两条边除了端点外，没有其他的交点．有些图确实能这样，凡能够这样画出的图叫做平面图．例如，图 1.1(*b*) 中的图 G 就是一个平面图．图 1.2(*a*) 中的图，本来是由一个空间六面体的顶点和棱组成的，但是它也可以画成图 1.4 中的那个样子（注意，三个要素一点也没有变）．因此，它也是一个平面

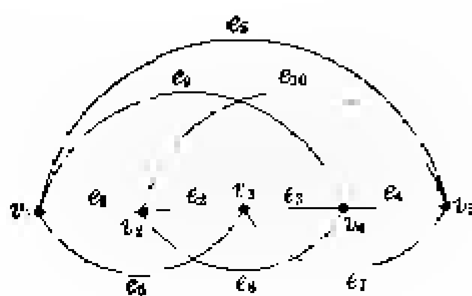


图 1.3

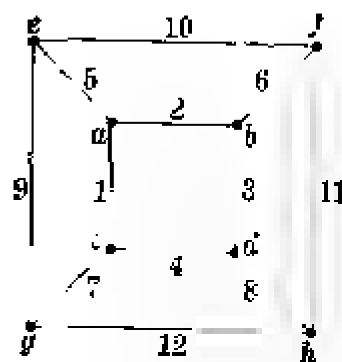


图 1.4

图．可是也有些图，不管怎样画，都不可能把所有顶点和边画在一个平面上，而使得任意两条边除了端点以外没有其他交点，这种图就叫做非平面图．图 1.2(b)或图 1.3 中的图 G_2 就是这样的一个例子．大家可以试试看，有没有办法把它的 5 个顶点和 10 条边都画在平面上．例如图 1.5 中已经画了 G_2 的 9 条边，还差一条连接 v_3 和 v_5 的 e_7．这条边不管怎么画，总要和其他边相交(当然，试了几次，画不出来，还不能就此断定它是非平面图．要肯定它是非平面图是需要严格证明的，因为证明较麻烦，这里就不仔细讲了)．

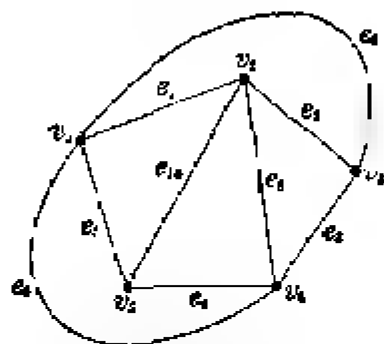


图 1.5

习　　题

1．写出图 1.1(b)、1.2(a) 中的无向图的三个要素．

2．已知图 $G=[V, E]$ 的三个要素为：

(1) $V=\{a, b, c, d, e, f\}$，

(2) $E=\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$，

(3) 每条边的端点如下表所示．

边	端点	边	端点
1	a, b	7	[illegible], d
2	b, c	8	b, d
3	c, d	9	f, c
4	d, e	10	c, f
5	c, e	11	c, d
6	a, e		

试把 G 画出来，G 是不是平面图？

2. 已知一个无向图 G 的顶点集合为 $V=\{v_1, v_2, v_3, v_4\}$，还知道 v_1 与 v_3，v_1 与 v_4，v_2 与 v_3，v_2 与 v_4 之间都有边相连，其他顶点间没有边相连．根据这些条件，是否已经可以把图 G 画出来了？

3．无向图的“抽象”定义

在上节的定义 1.1 中，我们给无向图下了一个定义．不过，很多书上给无向图下的定义和定义 1.1 不一样．下面的定义 1.2 就是常见的一种．

定义 1.2 设 $V=\{v_1, v_2, \cdots, v_n\}$ 及 $E=\{e_1, e_2, \cdots, e_m\}$ 是任意两个有限集合，并且满足：

(1) V 不是空集合；

(2) 每一个 $e_j \in E$ 是 V 的一个无序元素对 $[v_s, v_t]$，并且 $v_s \neq v_t$．

则称 V 和 E 这一对集合组成了一个无向图 G，记作 $G=[V, E]$．V 的元素 $v_1, v_2, \cdots, v_n$ 称为 G 的顶点，E 的元素 $e_1, e_2, \cdots, e_m$ 称为 G 的边．

定义 1.2 要比定义 1.1 抽象些了．因为在定义 1.1 中，集合 V 和 E 包含的东西很明确，V 的元素 v_i 是空间中的点，E 的元素 e_j 是线．而在定义 1.2 中，V 是由什么组成的，根本没有说．而 E 的元素又是什么 V 的“无序元素对”，好象也难以捉摸．

让我们来仔细分析一下，定义 1.2 是怎样产生的．上一节讲无向图的三个要素时，已经说过，对于一个图来说，最本质的东西是：

(1) 顶点集合 V；　　(2) 边集合 E；

(3) 每一个 $e_j \in E$ 以 V 中哪两个顶点为端点.

至于点的位置如何分布，线是直的还是弯的，都无关紧要. 其实，再进一步想想，如果 V 的元素不是空间中的点，而是别的东西，例如城市、村庄、驾驶员等；E 的元素不是线，而是公路、电线、"可以同机飞行"等. 只要 E 的每一个元素 e_j 都在某种意义下连接了 V 的两个元素，那末把 V 和 E 放在一起称为一个无向图又有什么不可以呢？表面上看，这种图和定义 1.1 中的图不一样，因为它的顶点不一定是空间中的点，而可以是任意的东西，但是，从本质上看，或者说，从它们都具有三个要素这一点来看，却是一致的. 而且定义 1.2 适应面更广.

定义 1.2 就是从这样的想法中产生出来的.

再来讲一下，为什么说 E 的元素 e_j 是 V 的无序元素对 $[v_s, v_t]$. 这种说法，对于初学的人，可能不大习惯. 其实，仔细想想也不陌生，在平面几何中不是也常常把一个以 A、B 为端点的线段叫做线段 AB 吗？这也就是把一个线段看成一对点. 在无向图中，对于一条边 e_j 来说，它究竟是什么东西是无关紧要的，主要的是它把哪两个点连接起来了，因此，完全可以把一条边 e_j 用它的端点 v_s 与 v_t 来表示，或者干脆就把边 e_j 看成是一对顶点 v_s 与 v_t，当然 v_s 与 v_t 谁在前谁在后都可以，因此，可以把 e_j 看成 V 的无序元素对 $[v_s, v_t]$.

再来看看引言中讲的那个飞机驾驶员问题，就会更清楚地看出把 e_j 看作 $[v_s, v_t]$ 是很自然的事了. 在那个问题中，有一个由 10 个驾驶员组成的集合 $V=\{v_1, v_2, \cdots, v_{10}\}$. 另外，如果一对一对地来考察这些驾驶员，那么有些"对"能够同机飞行，有些"对"不能. 我们可以这样想象：在那些能同机飞

行的驾驶员"对"之间，有一条无形的绳索把他们联系着，这些无形的绳索的集合就是边集合 E. 但是，如果我们追问一句，到底 E 的元素是什么？仔细想想，恐怕最恰当的回答就是："E 的元素就是 V 的一对元素 $[v_i, v_j]$" 了吧. 另外，我们也可以看出，对于一个无向图来说，只要知道了顶点集合 V，以及知道了哪些顶点对 v_i 与 v_j 间有边相连，这个图就完全知道了(回想一下上节的习题3).

一般，我们把用定义 1.2 定义的无向图叫做抽象的无向图，因为它的顶点集合是由任意的东西组成的，而把用定义 1.1 定义的叫做几何的无向图，因为它的顶点集合和边集合是由空间中的点和线组成的. 那末，今后提到无向图 $G=[V, E]$ 时，究竟指的是抽象的还是几何的呢？回答是，一般说来，指的是抽象的图，但是，我们可以把它看作是几何的图. 这是因为，虽然我们遇到的图的顶点可以代表各种不同的对象，例如城市、驾驶员等等，因而它是抽象的. 但是，对于任意一个抽象的无向图，我们总可以画一个几何的无向图来代表它. 例如有一个图 $G=[V, E]$，其中 V 是5个人的集合，$V=\{王, 丁, 张, 李, 陈\}$，$E=\{e_1, e_2, \cdots, e_8\}$ 并且 $e_1=[王, 李]$，$e_2=[王, 陈]$，$e_3=[王, 张]$，$e_4=[张, 李]$，$e_5=[张, 陈]$，$e_6=[张, 丁]$，$e_7=[李, 陈]$，$e_8=[李, 丁]$(注意，这里 G 是抽象的无向图). 那末，我们就可以画出象图 1.6 这样的一张图(图 1.6 是几何的图)来代表 G. 因此，虽然我们研究的是抽象的图，我们却可以把它们表示成一个具体

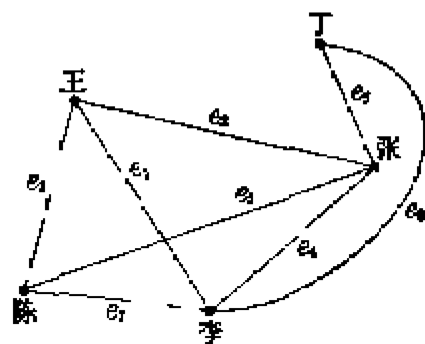


图 1.6

的、几何的图，这种表示，对于我们理解图论中的一些概念和以后学习图论中的各种极值问题的计算方法，都有很大的好处。

4. 路与回路

由于图论产生的时间还不长，因此，许多基本概念和定义还不统一，同一个名词，在不同的书上往往有不同的意义。因此，每一本讲图论的书，总要花一定的篇幅来把这本书中基本名词的意义解释一下，下面就来进行这一工作。

下面的定义都是对一个无向图 $G=[V, E]$ 来讲的，其中

$$V=\{v_1, v_2, \cdots, v_n\},$$
$$E=\{e_1, e_2, \cdots, e_m\}.$$

首先，如果边 e_j 以顶点 v_s 与 v_t 为端点，就记作 $e_j=[v_s, v_t]$。因为 $[v_s, v_t]$ 是无序元素对，因此，也可以写成 $e_j=[v_t, v_s]$（为简单起见，也可以记作 $e_j=[t, s]$）。

其次，如果顶点 v_s 是边 e_j 的一个端点，我们就说 v_s 与 e_j 是关联的，也可以说 e_j 与 v_s 是关联的。

定义 1.3 设两条边 e_i 与 e_j 具有相同的端点，即

$$e_i=[v_s, v_t],\ e_j=[v_s, v_t]$$

则称边 e_i 与 e_j 为平行边。没有平行边的无向图叫做简单图。有平行边的图叫多重图。

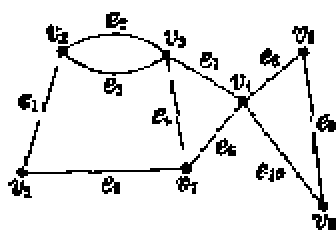


图 1.7

例如图 1.7 中的 e_2 与 e_3 是平行边，因此 G 是多重图，而前面图 1.2 中的三个图都是简单图。

定义 1.4 设 e_i 与 e_j 是图 G 的两条不相同的边，如果存在一个顶点 v_k，它是 e_i 与 e_j 的公共端点，就称 e_i 与 e_j 相邻。设 v_r 与 v_s 是 G 的两个顶点，如果存在一条边 e_k，它以 v_r 与 v_s 为端点，就称 v_r 与 v_s 相邻。

例如图 1.7 中，边 e_1 与 e_2 相邻，e_2 与 e_3 也相邻，而 e_1 与 e_6 则不相邻。另外顶点 v_4 与 v_5 相邻，而 v_2 与 v_4 就不相邻。

下面来介绍一个图论中很重要的概念——路。拿图 1.7 来说，假设它代表的是一个公路网，如果我们要从 v_1 走到 v_4 去，就可以按照下面的"路"走：从 v_1 出发，沿 e_1 到 v_2，再沿 e_2 到 v_3，最后沿 e_7 到 v_4。这条"路"可以简单地记成：

$$p_1=\{v_1, e_1, v_2, e_2, v_3, e_7, v_4\}$$

现在来分析一下"路" p_1 有什么特点。不难看出，首先，它是由一个顶点、一条边、然后再是一个顶点、一条边，…排列而成的。其次，对于路中的每一条边来说，排在它前面的顶点和后面的顶点恰好就是它的两个端点。下面的定义就是从路的上述特点概括出来的。

定义 1.5 图 G 中的一个由顶点和边组成的序列 p（序列的意思是：这些顶点与边在 p 中是排好了次序的，不能把它们的次序随便颠倒）

$$p=\{v_1', e_1', v_2', e_2', \cdots, v_{k-1}', e_{k-1}', v_k'\},$$

如果满足：
$$e_1'=[v_1', v_2'],$$

$$e_2'=[v_2', v_3'], \cdots, e_{k-1}'=[v_{k-1}', v_k'],$$

就称 p 是一条连接顶点 v_1' 和 v_k' 的路（这里 $v_1', v_2', \cdots, v_k'$ 是 V 中的顶点，$e_1', e_2', \cdots, e_{k-1}'$ 是 E 中的边）。v_1' 与 v_k' 叫做路 p 的两个端点。如果 $v_1'=v_k'$，就称 p 是一条回路。

例如图 1.7 中，除了刚才讲的 p_1 以外，

$p_2=\{v_5, e_8, v_4, e_7, v_3, e_4, v_7, e_6, v_4, e_{10}, v_6\}$,

$p_3=\{v_5, e_8, v_4, e_6, v_7, e_4, v_3, e_7, v_4, e_7, v_3, e_2, v_2\}$,

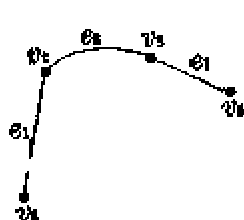
也都是路，而

$p_4=\{v_1, e_1, v_2, e_2, v_3, e_7, v_4, e_6, v_7, e_5, v_1\}$,

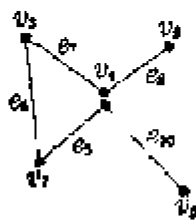
$p_5=\{v_3, e_4, v_7, e_6, v_4, e_{10}, v_6, e_9, v_5, e_8, v_4, e_7, v_3\}$,

$p_6=\{v_5, e_8, v_4, e_7, v_3, e_4, v_7, e_6, v_4, e_8, v_5\}$

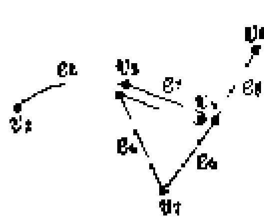
都是回路（见图 1.8）。



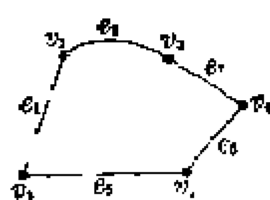
p_1 初级路



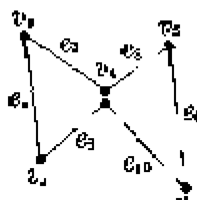
p_2 简单路



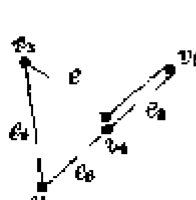
p_3 非简单路



p_4 圈或初级回路



p_5 简单回路



p_6 非简单回路

图 1.8

看一下p_1、p_2、p_3就可以看出它们之间还有区别，在p_1中，每一个顶点或边在路中至多出现一次；在p_2中，顶点v_4出现了二次，而每条边则至多出现一次；在p_3中，边e_7出现了二次（这时，当然e_7的端点也至少出现二次）。

定义 1.6 设p是一条路，

(1) 如果出现在p中的边都互相不同，则称p为**简单路**。

(2) 如果出现在p中的顶点都互相不同，则称p为**初级路**。

显然，图 1.8 中的 p_1 是初级路，p_2 是简单路（但不是初级路），而 p_3 不是简单路。

相似地，如果出现在一条回路 p 中的边都互不相同，就称 p 为简单回路，而当 p 中的顶点都互不相同时（除了 $v_1'=v_k'$ 以外），就称 p 为初级回路，或简称为圈。

例如图 1.8 中的 p_4 是圈，p_5 是简单回路，p_6 不是简单回路。

上面我们用

$$p=\{v_1', e_1', v_2', e_2', \cdots, v_{k-1}', e_{k-1}', v_k'\}$$

来表示一条路。也就是说，把出现在路中的所有顶点和边都写出来了。为了简单起见，也可以只把出现在路中的边写出来，例如图 1.8 中的 $p_1=\{v_1, e_1, v_2, e_2, v_3, e_7, v_4\}$ 也可以写成：

$$p_1=\{e_1, e_2, e_7\}.$$

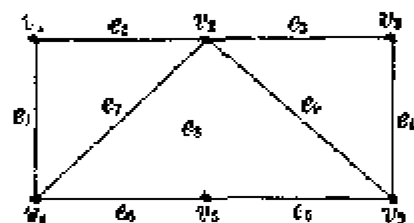


图 1.9

另外，如果考虑的图是简单图，那末也可以只写顶点。例如图 1.9 中的路：

$$p=\{v_1, e_2, v_2, e_8, v_5, e_5, v_6\}$$

就可以记成：

$$p=\{v_1, v_2, v_5, v_6\},$$

当然也可以记成：

$$p=\{e_2, e_8, e_5\}.$$

再强调一下，单用顶点的写法只适用于简单图，对于图 1.7 中的多重图来说，如果说到路 $p=\{v_1, v_2, v_3\}$，那末它的边既可能是 e_1 与 e_2 也可能是 e_1 与 e_3，这就有些混淆了。

习　题

1. 设 $p=\{v'_1, e'_1, v'_2, e'_2, \cdots, v'_{k-1}, e'_{k-1}, v'_k\}$ 是连接两个不同的顶点 v'_1 和 v'_k 的路，但不是初级路，证明一定可以从 p 中选出一部分边和顶点，组成一条连接 v'_1 和 v'_k 的初级路.

2. 举例说明下述命题是不成立的："从每一条非初级的回路中一定可以选出一部分边和顶点，组成一个圈".

3. 设 p 是图 $G=[V, E]$ 的一条回路，并且有一条边 e_i 在 p 中恰好出现一次，证明从 p 中一定可以选出一部分边和顶点组成一个圈.

5. 图的连通性，子图

定义 1.7 如果对于无向图 G 的任意两个不同的顶点 v_i 与 v_j，存在连接 v_i 与 v_j 的路(由上一节的习题 1，这时也存在连接 v_i 与 v_j 的初级路)，则 G 称为连通的.

看一下图 1.10 中的两个图，容易看出 (a) 是连通的，而 (b) 是不连通的，因为不存在连接顶点 a 与 e 的路.

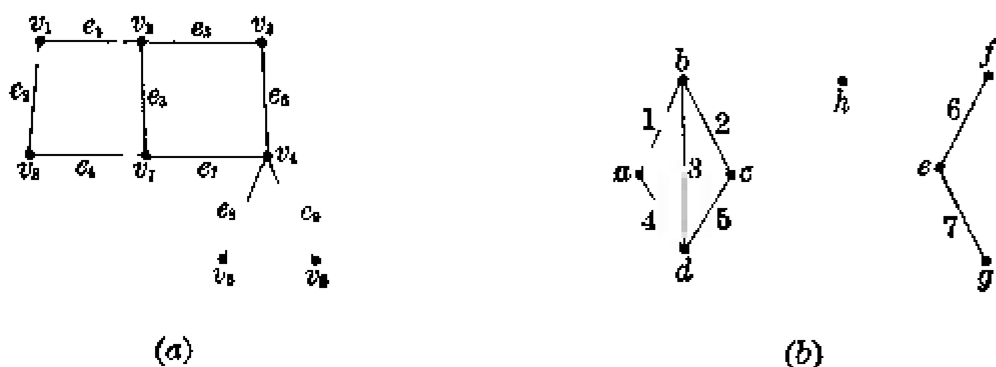


图 1.10

连通这个概念十分重要，它的直观意义就是"连成一片". 当然，如果把连通图定义为"连成一片的图"就不大好，因为如果再追问一句："什么叫连成一片？"恐怕就不好回答了. 而定

义 1.7 的意思是很明确的.

还有一个很重要的概念是子图. 在研究图论中的一些问题时, 往往需要研究一个已知图的一部分边和一部分顶点组成的图. 例如把全国的铁路网看成一个图 G, 那末一个省的铁路网就是由 G 的一部分边和顶点组成的图 G'. 一般, 我们有下述定义.

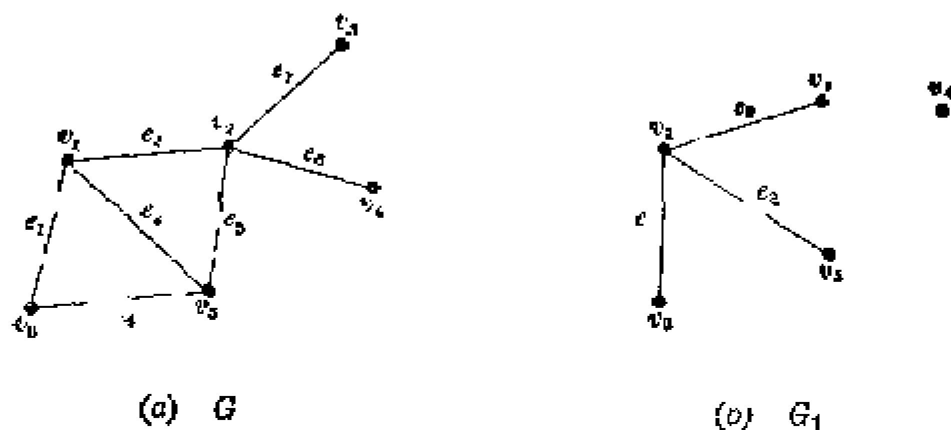
定义 1.8 设 $G=[V, E]$ 是一个无向图. V_1 与 E_1 分别是 V 与 E 的子集, 即 $V_1 \subseteq V$, $E_1 \subseteq E$, 如果对于任意 $e_i \in E_1$, e_i 的两个端点都属于 V_1, 则称 $G_1=[V_1, E_1]$ 是 G 的一个子图.

应注意: 并不是从 G 中任选一些边和顶点放在一起就组成 G 的一个子图 G_1. 只有满足: 如果某一条边被选入 G_1, 这条边的两个端点也被选入 G_1 时, G_1 才是一个子图.

例如图 1.11(b) 中的图 G_1 是 (a) 中的图 G 的子图. 而如果我们从 G 中选取边集合 $E_2=\{e_1, e_2, e_3\}$, 顶点集合

$$V_2=\{v_1, v_2, v_3, v_5\}$$

这时 V_2 与 E_2 合在一起就不是子图了. 因为边 e_1 的一个端点 v_6 不属于 V_2.



(a) G　　　　(b) G_1

图 1.11

定义1.9 设 $G_1=[V_1, E_1]$ 是 $G=[V, E]$ 的子图，并且 $V=V_1$，则称 G_1 是 G 的支撑子图.

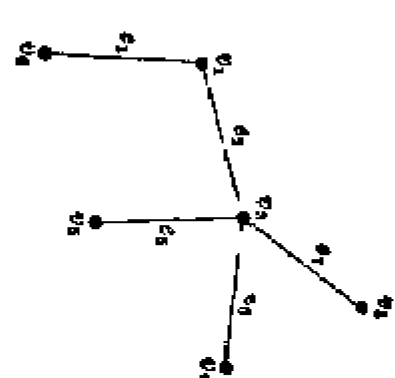


图 1.12

例如图1.12中的图 G_1 就是图1.11(a)中的图 G 的支撑子图.

也可以这样说：从一个图 G 中去掉一些边(顶点全部保留着)，剩下的就是一个支撑子图.

有两种形成子图的方法，以后经常要用到.

第一种办法是：设先给定了边集合 E 的一个子集合 E_1，我们再把属于 E_1 的所有边的端点都取出来，作成一个顶点子集合 V_1，显然 $G_1=[V_1, E_1]$ 是图 $G=[V, E]$ 的一个子图. 子图 G_1 叫做由边集合 E_1 生成的子图，记作 $[E_1]$.

第二种办法是：设先给定了顶点集合 V 的一个子集合 V_1，然后把 E 中所有两个端点都属于 V_1 的边都取出来，设这些边的集合为 E_1，显然 $G_1=[V_1, E_1]$ 也是 G 的一个子图，子图 G_1 叫做由顶点集合 V_1 生成的子图. 由 V_1 生成的子图记作 $[V_1]$.

例如在图1.11(a)的 G 中，如果取 $V_1=\{v_1, v_2, v_5, v_6\}$，则 V_1 生成的子图 $[V_1]$，就是图1.13中画的那个图. 如果取 $V_2=\{v_1, v_3, v_4\}$，则 V_2 生成的子图将只包含三个顶点，而不包含任何边，因为没有一条边的两个端点都属于 V_2.

再来看看图1.10(b)中的图 G，G 不是连通图. 但是不难看出，我们可以很自然地把 G 分成三个连通子图(见图1.14) G_1, G_2, G_3. 这种子图 G_1, G_2, G_3 都称为图 G 的连通分支.

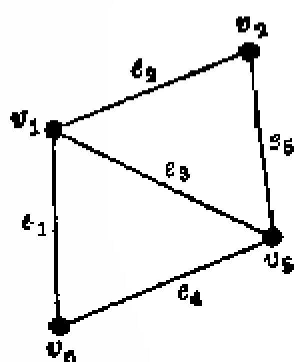


图 1.13

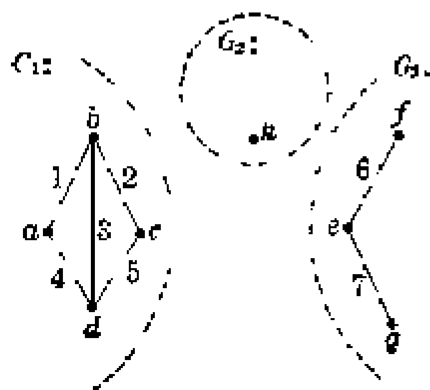


图 1.14

上面只是通过一个例子讲了什么叫做一个图的连通分支，下面再把这个概念比较仔细地讲一下。

设 v_i 与 v_j 是图 $G=[V, E]$ 的任意两个不同的顶点，那末可能存在一条以 v_i 与 v_j 为端点的路，也可能不存在以 v_i 与 v_j 为端点的路。因此，我们可以把顶点集合 V 中的所有顶点分成几个子集合，即把互相之间有路相连的顶点分在一个子集合中(以图 1.14 为例，顶点集合可以分成三个子集合 $\{a, b, c, d\}$, $\{e, f, g\}$, $\{h\}$)，然后按照前面讲的形成子图的第二种办法，让每一个这样的子集合生成一个子图，这样生成的子图就叫做图 G 的连通分支。

注意，如果一个图 G 是连通的，那末它只有一个连通分支。反过来，如果 G 只有一个连通分支，那末它一定是连通的。

习　　题

设 $G=[V, E]$为一个无向图，其中 $V=\{a, b, c, d, e, f, g, h, i\}$, $E=\{1, 2, \cdots, 8\}$, 并且 $1=[c, a]$, $2=[a, b]$, $3=[e, f]$, $4=[f, h]$, $5=[b, c]$, $6=[c, h]$, $7=[f, g]$, $8=[f, i]$.

(1) G 是不是连通的?

(2) G 有几个连通分支?

(3) 设 $V_1=\{a, c, d, f, g\}$，画出 V_1 生成的子图 $[V_1]$.

二、有 向 图

1、有向图的定义

图 2.1 中画出了一个图.它的每一条线上都有一个箭头，代表这条线的方向，这样的图叫做有向图.

和无向图一样，我们也可以给有向图下两个定义，一个是几何的，一个是抽象的.

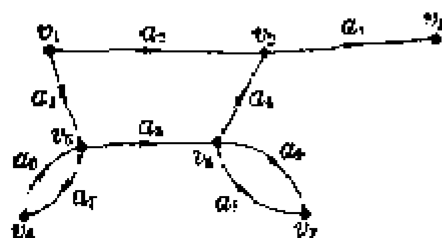


图 2.1

定义 2.1 设 $V=\{v_1, v_2, \cdots, v_n\}$ 是空间中 n 个点的集合，$A=\{a_1, a_2, \cdots a_m\}$ 是 m 条线的集合，如果满足：

(1) V 不是空集合.

(2) A 的每一条线 a_i 以 V 中的一个点 v_s 为起点而以另一个点 v_t 为终点(起点和终点统称为端点)；

(3) A 的任意两条线，除了起点或终点以外，没有其他的公共点，则把由 V 和 A 组成的图形叫做一个有向图，记作 $D=(V, A)$，V 中的点叫做 D 的顶点，A 中的线叫做 D 的弧.

注意：为了与无向图区别，这里 $D=(V, A)$ 用的是圆括号.

在学过了无向图以后，再来学有向图的这个"几何的"定

义，就比较容易了．它和定义 1.1 的唯一差别就是，在无向图中，每条线的两个端点的地位是一样的，而在有向图中，两个端点的地位就不一样了，其中有一个是起点而另一个是终点．为了把起点和终点在图上表示出来，只要在每一条线上画一个箭头，箭头指向这条线的终点，例如，图 2.1 中的 a_1，箭头指向 v_3，所以 a_1 的终点是 v_3，起点当然是 v_2 了．

有向图这一概念也是从生产实际中产生出来的．例如，研究渠道网时，在每一条渠道中，水只能沿着一个方向流动，即从高处向低处流．又如研究通讯系统，电报只能由发射台发到接受台，这种问题，如果归结成一个图，一般都是有向图．

和无向图一样，一个有向图也有三个要素，就是：

(1) 顶点集合 V；

(2) 弧集合 A；

(3) 每条弧以哪一个顶点为起点，以哪一个顶点为终点．

知道了这三点，一个有向图就可以认为是完全掌握了，并且和前面一样，也一定可以画出来了．这样，我们也可以给有向图下一个抽象的定义．

定义 2.2 设 $V=\{v_1, v_2, \cdots, v_n\}$ 及 $A=\{a_1, a_2, \cdots, a_m\}$ 是任意两个有限集合，并且满足：

(1) V 不是空集合；

(2) 每一个 $a_i \in A$ 是 V 的一个有序元素对 (v_s, v_t)，并且 $v_s \neq v_t$，则称 V 和 A 这一对集合组成了一个有向图，记作 $D=(V, E)$．

把这个定义与定义 1.2 比较一下，就会发现两者真是只有一字之差，就是条件 (2) 中的"无序元素对"改成了"有序元

素对”. 因此，这里就不必对这个定义多加说明了. 请大家结合上一章定义 1.1 与定义 1.2 自己来分析这两个定义之间的关系吧.

习　　题

1　写出图 2.1 中的有向图的两个要素.

2　已知有向图 $D=(V, A)$ 中，$V=\{v_1, v_2, \cdots, v_7\}$，$A=\{a_1, a_2, \cdots, a_9\}$，每条弧的起终点如表所示，试把 D 画出来.

弧	起　点	终　点	弧	起　点	终　点
a_1	v_1	v_2	a_6	v_4	v_5
a_2	v_2	v_2	a_7	v_1	v_4
a_3	v_2	v_1	a_8	v_4	v_6
a_4	v_4	v_6	a_9	v_1	v_6
a_5	v_6	v_5			

2. 有向图的路和有向路

和上一章一样，下面我们把有向图中的一些基本名词的意义解释一下：

设 $D=(V, A)$ 是一个有向图，其中 $V=\{v_1, v_2, \cdots, v_n\}$，$A=\{a_1, a_2, \cdots, a_m\}$. 如果弧 a_j 以 v_s 为起点，以 v_t 为终点，就记作 $a_j=(v_s, v_t)$（或 $a_j=(s, t)$）. 注意，这里用的是圆括号，也就是说，(v_s, v_t) 是有序元素对.

定义 2.3　设 a_i 与 a_j 是 D 的两条弧，并且有：

$$a_i=(v_s, v_t),\ a_j=(v_s, v_t),$$

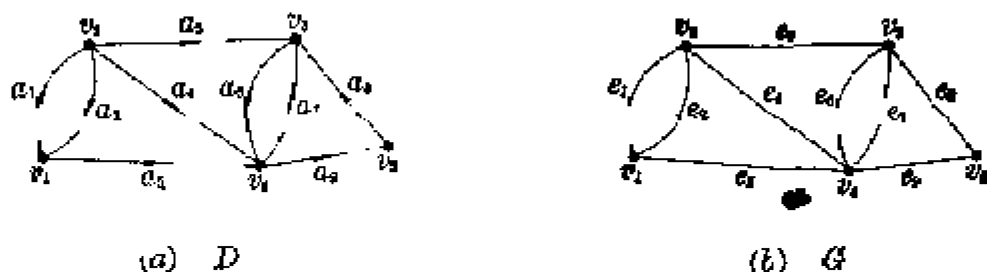
则称 a_i 与 a_j 为平行的，没有平行弧的有向图叫做简单有向图.

例如前面图 2.1 中的弧 a_8 与 a_9, 起点都是 v_5, 终点都是 v_7, 因此是平行弧. 注意, a_6 与 a_7 不是平行弧.

如果 $a_i=(v_s, v_t)$, 我们就说 a_i 是从 v_s 出发而指向 v_t 的一条弧.

定义 2.4 设 $D=(V, A)$ 是一个有向图. 如果把它的每一条弧的方向去掉, 即把每一条弧看成一条边(或者说把有序元素对 (v_s, v_t) 改成无序元素对 $[v_s, v_t]$), 可以得到一个无向图, 叫做 D 的相伴无向图.

例如图 2.2(a) 中的有向图 D 的相伴无向图就是 (b) 中的无向图 G. 注意, 虽然图 2.2(a) 中的 D 是简单的, 即它没有平行弧, 但它的相伴无向图却有平行边.



(a) D　　　　(b) G

图 2.2

下面来讨论一下, 与无向图中的路, 圈, 子图等等相似的一些概念.

首先, 子图和支撑子图的概念可以完全类似于无向图那样来定义, 这里不细讲了.

其次, 在有向图中, 要区别两种不同的路和回路.

定义 2.5 设 $D=(V, A)$ 是一个有向图,

$$p=\{v_1', a_1', v_2', a_2', \cdots, v_{k-1}', a_{k-1}', v_k'\}$$

是一个由 D 的顶点和边组成的序列,

1. 如果有: $a_1'=(v_1', v_2')$, $a_2'=(v_2', v_3')$, $\cdots$, $a_{k-1}'=(v_{k-1}',$

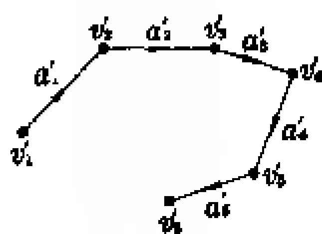
v_k')，就称 p 是一条从 v_1' 到 v_k' 的有向路，如果 $v_1'=v_k'$，则称 p 是一条有向回路。

2．如果有：$a_1'=(v_1', v_2')$ 或 (v_2', v_1')，

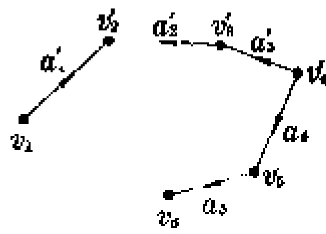
$$a_2'=(v_2', v_3') \text{ 或 } (v_3', v_2'),$$
$$\vdots$$
$$a_{k-1}'=(v_{k-1}', v_k') \text{ 或 } (v_k', v_{k-1}');$$

就称 p 是一条路，如果 $v_1'=v_k'$，则称 p 是一条回路。

图 2.3(*a*) 与 (*b*) 分别是有向路与路的示意图。注意：在有向图中同时有路与有向路的概念。有向路一定也是路，但是，反过来不对，即路不一定是有向路。



(*a*) 有向路



(*b*) 路

图 2.3

例如在图 2.4(*a*) 的 D 中，

$$p_1=\{v_1,\ a_1,\ v_2,\ a_2,\ v_3,\ a_8,\ v_4\}$$

是一条有向路；

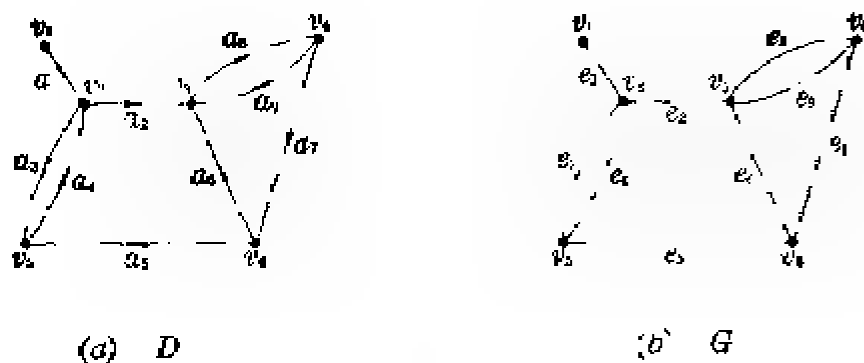
$$p_2=\{v_1,\ a_1,\ v_2,\ a_4,\ v_5,\ a_5,\ v_6,\ a_6,\ v_3\}$$

是路而不是有向路，

$$p_3=\{v_4,\ a_7,\ v_6,\ a_6,\ v_3,\ a_8,\ v_4\}$$

是一条有向回路。

有向路这一概念的实际意义是很明显的，例如我们把图 2.4(*a*) 中的 D 看成一个公路网，每条弧代表一条只可以单向行车的路，那末，沿着有向路 p_1 就可以从 v_1 走到 v_4，而沿着路 p_2 是不能从 v_1 走到 v_3 的。



(a) D　　　(b) G

图 2.4

让我们对有向图中的路这一个概念作一些进一步的分析。例如刚才说了，在图2.4(a)的D中，p_2是一条路(但不是有向路)，如果把p_2中的弧分别换成它们在D的相伴无向图G(见图2.4(b))中对应的边，即将a_1换成e_1，a_4换成e_4，…，这时，p_2就变成：

$$p_2' = \{v_1,\ e_1,\ v_2,\ e_4,\ v_5,\ e_5,\ v_6,\ e_6,\ v_3\}$$

很显然，p_2'是G的一条路。反过来，假设先有G的任意一条路，如果把这条路中的边都换成D中与它们对应的弧，就也可以得到D的一条路。因此，我们也可以把有向图的路定义成：

设：　$p = \{v_1',\ a_1',\ v_2',\ a_2',\ \cdots,\ v_{k-1}',\ a_{k-1}',\ v_k'\}$，

是有向图$D=(V,\ A)$中的一个由顶点和弧组成的序列，它具有下述性质：如果不看弧的方向(即把a_i'换成相伴无向图中的e_i')，p就是相伴无向图中的一条路，则称p为D的一条路。

上面对路的定义的分析，主要是想说明一个问题，就是对于有向图来说，有许多概念可以通过它的相伴无向图来定义，例如路就是。这种概念本质上还是属于无向图的。但是也有一些概念却是有向图所特有的，例如有向路就是，这一观点下

面常常要用到.

在有向图中，也可以定义简单路，简单有向路，初级有向路，初级有向回路(即有向圈)等，这些都不细讲了，请大家对照着无向图的情况考虑这些概念应该怎样定义. 另外，和无向图中相似，在写有向路时，可以只写在向路中出现的弧，而当考虑简单图时，也可以只写顶点.

习 题

1. 考虑一下，在有向图中应该如何给子图，支撑子图，简单有向路，初级有向路等概念下定义.

2. 证明：设 p 是从 v_i 到 v_j 的有向路，但不是初级有向路，则一定可以从 p 中选出一部分顶点和弧，组成一条从 v_i 到 v_j 的初级有向路.

3. 强连通性

在有向图中，也要考虑两种不同的连通性.

定义 2.6 如果有向图 $D=(V, A)$ 的相伴无向图是连通的，就称 D 是**连通的**.

显然，这又是一个利用相伴无向图来定义的概念，例如图 2.5(a) 中的有向图是连通的，而 (b) 中的有向图就是不连通的.

连通分支概念也可以利用相伴无向图来定义，这里就不仔细讲了. 例如图 2.5(b) 中的图有三个连通分支.

更加重要的是一种与方向性有密切关系的连通性，叫做强连通性，在讲强连通性以前，先介绍另一个重要的概念.

定义 2.7 设 v_i 与 v_j 是有向图 $D=(V, A)$ 的两个顶点，如果存在一条从 v_i 到 v_j 的有向路，就说由 v_i 可达 v_j. 如果

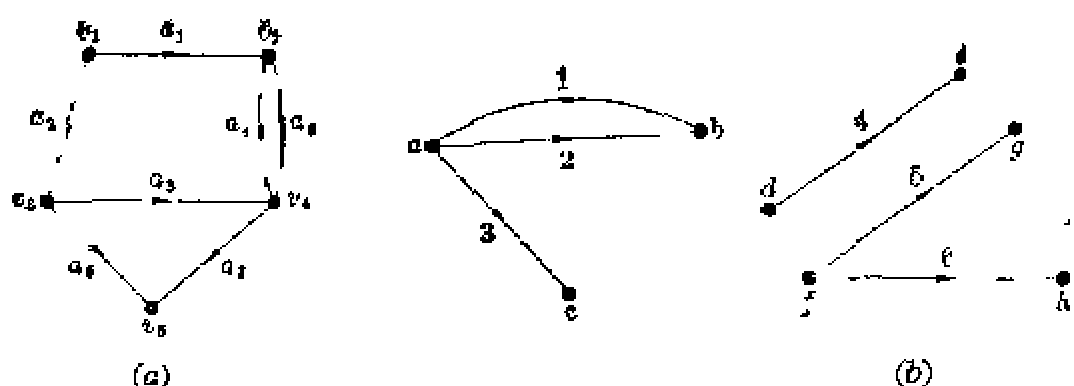


图 2.5

由 v_i 可达 v_j，同时由 v_j 可达 v_i，则称顶点 v_i 与 v_j 是互相可达的．此外我们也认为每一个顶点 v_i 可达 v_i 本身．

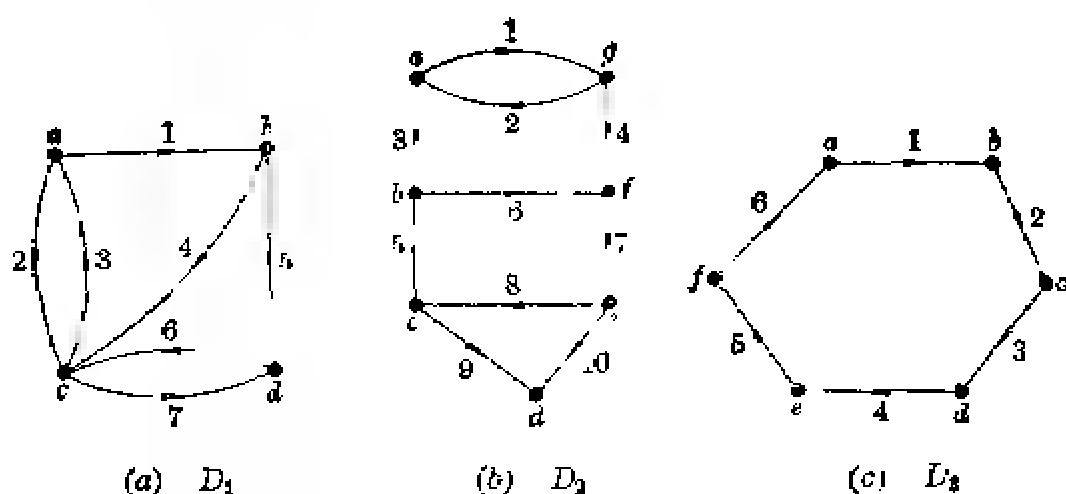
例如在图 2.5(a) 的有向图中，由 v_1 可达图中所有的顶点，由 v_2 可达 v_4、v_5 及 v_2 本身，而由 v_5 只可达 v_5 自己．如果这个图代表的是一个渠道网，而且水只能沿着箭头的方向流动，那末如果由顶点 v_i 可达顶点 v_j，水就能够从 v_i 流到 v_j．

定义 2.8 设 $D=(V, A)$ 是一个有向图，如果 D 的任意两个不同的顶点 v_i 与 v_j 都是互相可达的，则称 D 是一个强连通图．

定义 2.8 也可以说成是，如果对于有向图 D 的任意两个不同的顶点 v_i 与 v_j，既存在从 v_i 到 v_j 的有向路，也存在从 v_j 到 v_i 的有向路，就称 D 是强连通的．

例如图 2.6(a)(c) 中的 D_1 与 D_3 都是强连通的(请大家自己检查一下)，而图 2.6(b) 中的 D_2 就不是强连通的(注意，它是连通的)，因为不存在从 b 到 a 的有向路．

与连通分支概念类似，在有向图中可以研究强连通分支．拿图 2.6(b) 中的 D_2 来说，它不是强连通的，但是，我们可以从图中找出一些强连通的子图来，办法和上一章中找连通分



(a) D_1　　(b) D_2　　(c) D_3

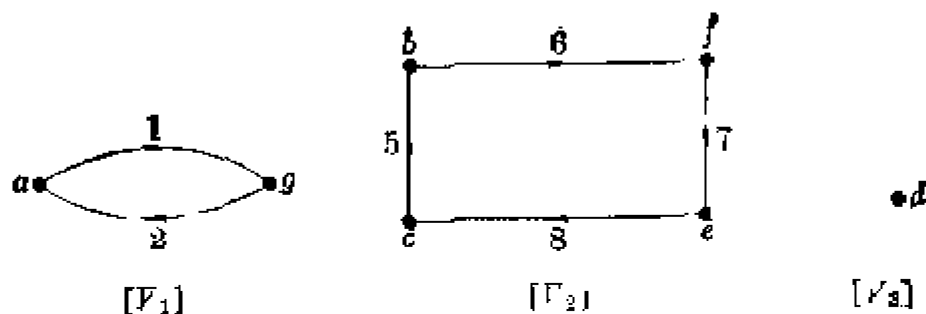
图 2.6

支类似，就是：首先，把所有顶点分分类，即把顶点集合分成几个子集合，把互相可达的顶点都分在同一个子集合中．例如，先考虑顶点 a，看看有哪些顶点与 a 互相可达，不难看出，只有 g 与 a 互相可达，因此 $\{a, g\}$ 是一个子集合；再取一个不属于上述子集合的顶点，例如取顶点 b，b 与 c、b 与 e、b 与 f 都互相可达（这时，c 与 e，c 与 f，等等一定也都互相可达．想想看，为什么？），因此，又可以得到一个子集合 $\{b, c, e, f\}$，还剩下一个顶点 d 了，不难看出，d 与任何别的顶点都不互相可达，因此，d 一个顶点就构成一个子集合 $\{d\}$，这样，我们就把 D_2 的顶点集合分成三个两两没有公共顶点的子集合：

$$V_1=\{a, g\},\ V_2=\{b, c, e, f\},\ V_3=\{d\}.$$

然后，让每一个上述的这些子集合生成一个 D 的子图．例如 V_1 生成的子图 $[V_1]$ 包含顶点 a, g 以及起点和终点都属于 V_1 的所有弧，这里就是弧 1 和 2，同理可以得到 $[V_2]$，$[V_3]$（见图 2.7），这样得到的子图 $[V_1]$，$[V_2]$，$[V_3]$ 就叫做原来有向图的强连通分支．

上面通过一个例子说明了什么叫强连通分支，一般情况



[Γ_1]　　[Γ_2]　　[Γ_3]

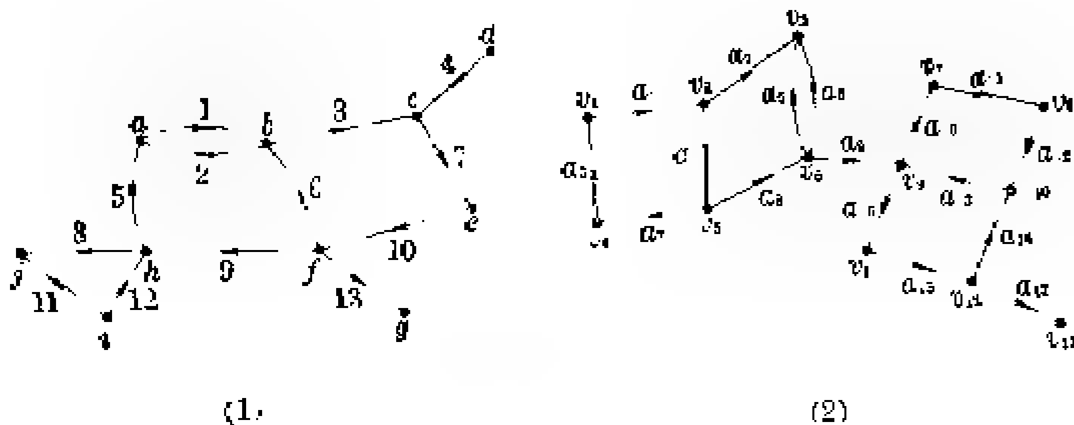
图 2.7

也是完全相似的，这里就不细讲了.

应该注意，无向图中讲的连通分支与这里的强连通分支有一个很大的区别，就是，在无向图中，每一条边至少属于一个连通分支，而现在，却并不是每一条弧都属于一个强连通分支，例如，图 2.6(b) 中的弧 3, 4, 9, 10 就不属于任何强连通分支.

习　　题

找出下面两个有向图的所有强连通分支，指出哪些弧不属于任何强连通分支.



(1)　　(2)

4. 对称有向图，有向图与无向图的关系

上一章与这一章分别讲了无向图与有向图的定义和一些

基本概念．最后再来讨论一下这两种图之间的关系．为此，先来讨论一下什么叫做对称有向图．

定义 2.9 设 $D=(V,A)$ 为一个有向图，a_i 与 a_j 是 D 的两条弧，如果有：

$$a_i=(v_s,v_t),\quad a_j=(v_t,v_s),$$

则称 a_j 是 a_i 的反向弧（当然 a_i 也是 a_j 的反向弧）．

定义 2.10 设 $D=(V,A)$ 是一个简单有向图（即没有平行弧的有向图），如果 D 的每一条弧都有反向弧，就称 D 是一个对称的有向图．

对称有向图的特点是，两个顶点之间如果有弧相连，一定是两条，并且方向相反（不会多于两条，否则就不是简单有向图了）．例如图 2.8(*a*) 中画的就是一个对称有向图．这种图的重要之处在于，我们可以把一个对称有向图看成一个无向图，即把两条互为反向弧的弧看成一条无向边（见图 2.8(*b*)）．这种做法是有实际意义的，例如当我们用图来代表一个公路网时，这时图上的无向边代表一条两个方向都能行驶车辆的路，而有向弧则代表只能单向行驶的路，因此当一条有向弧有反向弧时，就表示这一段路上朝两个方向都能走，所以就相当于一条双向可行的路．

反过来，我们也可以把一个无向图看成一个对称的有向



(*a*)　　　　(*b*)

图 2.8

图，即把每一条无向边看成两条方向相反的有向弧。

由此可以得出这样的结论："每一个无向图都可以看成是一个有向图"。办法就是把每一条无向边都看成两条有向弧。这么一说，大家可能会产生这样的印象：用不到研究无向图，只要研究有向图就够了。这种看法倒也未必正确。因为一方面，把一个无向图中无向边都画成两条弧来研究，有时不如直接研究无向图来得方便；另一方面，在图论中，有些概念从本质上来说是与方向无关的（这一点前面已经多次提到过了），把这种概念，以及与这些概念有关的定理等等，作为无向图的内容来研究，一般来说，常常会显得更自然些。

有时，在遇到象图 2.9(a) 中画的那种不对称的有向图时，为了方便，我们把两个顶点之间的一对方向相反的弧约简为一条无向边，这样就得到一个既有有向弧，又有无向边的"混合图"了（见图 2.9(b)）。这里强调说一下，以后凡遇到混合图时，我们都把它理解成本来是一个有向图，只是为了方便，把它的若干对有向弧画成无向边了。

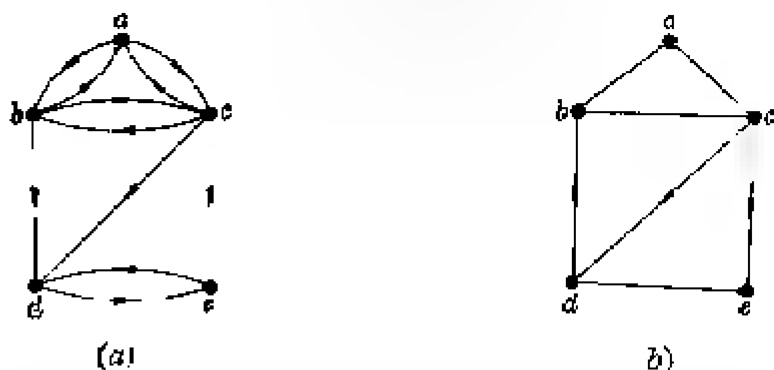


图 2.9

有关图论的基本知识，就暂时讲到这里。下一章开始转入这本书的主题，就是讲图论中的各种极值问题和它们的解法。在讲各种极值问题时，还需要用到另外一些图论中的概念和定理，这些就分散在以后的各章中随用随讲了。

三、最短路问题

1. 什么是最短路问题

从这一章开始，每一章讲一个图论中的极值问题. 这一章先讲一个比较简单但又是很重要的问题，就是最短路问题.

让我们先把最短路问题的提法明确一下：

1. 求有向图上的最短有向路问题. 设 $G=(V, A)$ 是一个有向图，它的每一条弧 a_i 都有一个非负的长度 $l(a_i)$。在 G 中指定了两个顶点 v_s 与 v_t，要求把从 v_s 到 v_t 并且长度最小的有向路找出来.

2. 求无向图上的最短(无向)路问题. 设 $G=[V, E]$ 是一个无向图，它的每一条边 e_i 都有一个非负的长度 $l(e_i)$，在 G 中指定了两个顶点 v_s 与 v_t，要求把连接 v_s 与 v_t 并且长度最小的(无向)路找出来.

上面两个问题都可以称为最短路问题. 很容易看出，这两个问题都有着大量的生产实际背景. 事实上，大至海、陆、空各种运输，小至一个人每天上班，都会遇到最短路问题. 正因为它用处大，所以近二、三十年来国内外对这个问题进行了不少研究，也找到了许多比较好的计算方法.

有趣的是，有些问题，从表面上看与最短路问题没有什么

关系，却也可以归结为最短路问题．下面就来举一个这样的例子．

〔例1〕 渡河问题．一个人带了一只狼、一只羊和一棵白菜想要过河，河上有一只独木船，每次除了人以外，只能带一样东西．另外，如果人不在旁时，狼就要吃羊，羊就要吃白菜．问应该怎样安排渡河，才能做到既把所有东西都带过河去，而且在河上来回的次数又最少？

当然，这个问题不用图论也能解决．大家一眼就能看出，第一次应该带着羊过河，让狼和白菜留下，以下怎么渡法，请大家考虑．

下面就来讨论一下，怎样把这个问题转化成最短路问题．

我们用 M 代表人，W 代表狼，S 代表羊，V 代表白菜．开始时，设人和其他三样东西都在河的左岸，这种情况，我们用 $MWSV$ 来表示．又例如人带了羊渡到河的右岸去了，这时左岸留下了狼和白菜，这种情况就用 WV 来表示．大家想想看 MWS 表示什么？按照前面讲的，应该表示人（M）、狼（W）、羊（S）在左岸而白菜（V）在右岸这种情况．那末总共可能有几种允许的情况呢？如果不管狼是否吃羊、羊是否吃白菜，那末总共有 16 种情况，它们分别是：

$MWSV$， MWS， MWV， MSV，

WSV， MW， MS， MV，

WS， WV， SV， M，

W， S， V， ϕ（空集）

例如 MS 表示人和羊在左岸，而狼和白菜在右岸；ϕ 表示左岸是空集，即人、狼、羊、白菜都已渡到右岸去了．检查一下就可以知道，这 16 种情况中有 6 种情况是不允许出现的．就

是 WSV, MW, MV, WS, SV, M. 例如 WSV 表示狼、羊、白菜都在左岸而人在右岸，因为人不在旁边看着，狼就要吃羊，羊也要吃白菜；又如 MV 表示人和白菜在左岸，而狼和羊在右岸，当然也是不行的。因此，允许出现的情况只有 10 种。

现在我们就来构造一个图 G（见图 3.1），它的顶点就是这 10 种情况，G 中的边是按照下述原则来连的：如果情况甲经过一次渡河可以变成情况乙，那末就在情况甲与乙之间连一条边。

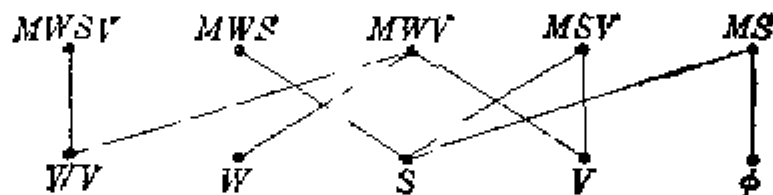


图 3.1

例如，$MWSV$ 经过一次渡河可以变成 WV（人带着羊过河，左岸留下狼和白菜），又例如 MWV 经过一次渡河可以变为 W（人带着白菜过河，留下狼），或变为 V. 当然反过来，W 也可以变为 MWV（人带着白菜从右岸返回左岸）。

作出了图 G 以后，渡河问题就归结为下述问题了："在图 G 中找一条连接顶点 $MWSV$ 与 ϕ，并且包含边数最少的路"。如果我们设 G 的每条边的长度都是 1，那末也可以把渡河问题归结为"找一条连接 $MWSV$ 与 ϕ 的最短路"。

请大家试试，在图 3.1 中找一条连接 $MWSV$ 与 ϕ 的最短路，并且按照找出的路来给出渡河问题的一个答案。

大家也许会认为，这个渡河问题本来就不很难，把它转化成图论问题，倒相当麻烦，有什么好处呢？其实这种做法还是

很有好处的．因为在转化前，想解决渡河问题，只能用凑的办法，或者最多是凭经验．而转化成图论问题以后，就可以用一种系统的方法来解决了．

最后，还要指出一下，求最短有向路和求最短无向路这两个问题是密切关联的．下面将看到，求最短有向路的计算方法也可以用来求最短无向路．因此，我们将在下一小节中先介绍最短有向路的求法，以后再讲无向图上最短路的求法．

另外，在这一章中，我们假设遇到的图 G 都是简单图．这样假设是合理的，因为如果 G 有平行弧或平行边，例如有好几条从 v_i 到 v_j 的弧，那末很显然，可以把这些弧中最短的一条留下，其余的都去掉．然后在剩下的简单图上再来求从 v_s 到 v_t 的最短有向路．因为 G 是简单图，所以每一条弧 a_k 被它的起点 v_i 与终点 v_j 唯一决定，因此，下面我们就用 (v_i, v_j) 或 (i, j) 来表示一条弧，用 $[v_i, v_j]$ 或 $[i, j]$ 来表示边，而用 $l(i, j)$ 来表示弧或边的长度．

2．求最短有向路的标号法

这一节介绍一种求有向图上最短有向路的方法，叫做标号法．所谓标号，我们是指与图的每一个顶点对应的一个数（或几个数）．例如设 $G=(V, A)$ 的顶点集合是 $V=\{v_1, v_2, \cdots, v_n\}$，如果我们能使 v_1 对应一个数 $b(1)$，v_2 对应数 $b(2)$，$\cdots$，v_n 对应数 $b(n)$，那末，这些数 $b(i)$ 就称为 v_i 的标号．当然，在不同的问题中，标号 $b(i)$ 一般代表不同的意义．

回到我们要解决的最短有向路问题上来．为确定起见，我们设 $v_s=v_1$，$v_t=v_n$，也就是说我们要找的是从 v_1 到 v_n 的

最短有向路．下面介绍的方法可以把从 v_1 到 G 的每一个顶点 v_j 的最短有向路都求出来（或者指出不存在从 v_1 到 v_j 的有向路，即 v_1 不可达 v_j）．

我们把整个计算分成若干"轮"来进行（一个"轮"就是一个大步），每一轮中，将求出 v_1 到某一个顶点 v_j 的最短路以及这条最短路的长度 $b(j)$．我们把 $b(j)$ 就叫做顶点 v_j 的标号．再强调一下，请大家记住，顶点 v_j 的标号代表的是 v_1 到 v_j 的最短路的长度．另外，如果说起"顶点 v_j 已经有标号了"或"v_j 是已标号点"，就意味着从 v_1 到 v_j 的最短路以及这条最短路的长度都已经求出来了．

计算开始时，令 $b(1)=0$，v_1 变为已标号点，其余顶点都是未标号点．这样做的意义很清楚，因为 $b(1)$ 代表从 v_1 到 v_1 的最短路的长度，当然不用计算就可以知道，它应该等于 0．

如果计算是在一张图上进行，那末我们可以在顶点 v_1 旁边写一个数 0，并用较深的颜色写，表示这是 v_1 的标号并且已算出．例如要求下面图 3.2 的图 G 中从 v_1 到 v_7 的最短有向路，那末在开始计算时应该在 v_1 旁边写一个 **0**．

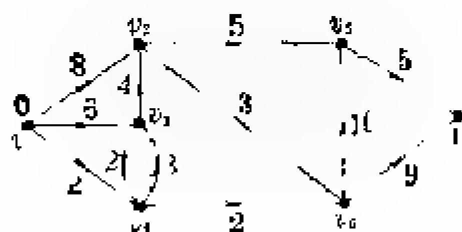


图 3.2

每一轮计算可以分成下面几个步骤．

步骤 1　找出所有具有下述性质的弧 (i, j)：起点 v_i 是已标号点而终点 v_j 是未标号点．如果这样的弧不存在，计算结束．

以图 3.2 为例，在进行第一轮计算时，这样的弧共有三

条，就是 (1, 2)、(1, 3)、(1, 4)。

步骤 2 对于步骤 1 中找到的每一条弧 (i, j)，计算一个数：

$$k(i, j)=b(i)+l(i, j)$$

（如果这个 $k(i, j)$ 在前面各轮计算中已经算过，就不必再算）也就是说：$k(i, j)$ 等于弧 (i, j) 的起点的标号加上弧的长度。把算出的 $k(i, j)$ 的值就写在弧 (i, j) 的旁边，并在数的外面加上一个方括号。然后找出使 $k(i, j)$ 最小的弧 (c, d)（如果有好几条弧都使 $k(i, j)$ 达到最小，可任取一条）。

在图 3.2 中：

$$k(1, 2)=b(1)+l(1, 2)=0+8=8,$$
$$k(1, 3)=b(1)+l(1, 3)=0+6=6,$$
$$k(1, 4)=b(1)+l(1, 4)=0+2=2,$$

使 $k(i, j)$ 最小的弧是 (1, 4)。

步骤 3 把弧 (c, d) 画成粗线，把顶点 v_d 变为已标号点，令 v_d 的标号 $b(d)$ 就等于 $k(c, d)$。这一轮计算结束。

在图 3.2 中，应将弧 (1, 4) 加粗，顶点 v_4 得到标号 $b(4)=k(1, 4)=2$（见图3.3）。

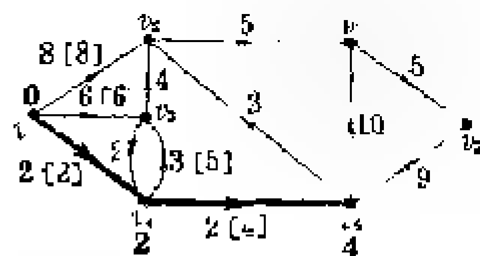


图 3.3

在一轮计算结束后，应该检查一下，是不是所有顶点都得到标号了，如果是的，那末整个计算就结束了；如果不是，即还有未标号的顶点，就转向下一轮计算（即再从步骤 1 开始计算）。

例如对图 3.3 继续进行第二轮计算如下。

步骤 1 有 4 条弧具有"起点已标号，终点未标号"这一

性质，它们是 $(1, 2)$，$(1, 3)$，$(4, 3)$，$(4, 6)$。

步骤2　计算出：

$$k(4, 3)=2+3=5,$$

$$k(4, 6)=2+2=4,$$

$k(1, 2)=8$ 与 $k(1, 3)=6$ 在前一轮中已经标出，并已记在弧的旁边了，故不必再算。最小的是 $k(4, 6)=4$，所以使 $k(i, j)$ 最小的弧是 $(4, 6)$。

步骤3　把弧 $(4, 6)$ 加粗，v_6 成为已标号点，

$$b(6)=k(4, 6)=4.$$

也就是说，v_1 到 v_6 的最短有向路已经找到了，就是 $p=\{v_1, v_4, v_6\}$，它的长度是4(见图3.3)。

在第三轮计算中，因为已标号点有 v_1, v_4, v_6 三个，步骤1中考虑的弧将有6条，即 $(1, 2)$，$(1, 3)$，$(4, 3)$，$(6, 2)$，$(6, 5)$，$(6, 7)$。对应的 $k(1, 2)=8$, $k(1, 3)=6$, $k(4, 3)=5$, $k(6, 2)=7$, $k(6, 5)=14$, $k(6, 7)=13$。最小的是 $k(4, 3)=5$，因此应该把弧 $(4, 3)$ 加粗，并且让顶点 v_3 得到标号 $b(3)=5$。

继续做下去，在进行了六轮计算后(请读者自己做一遍)，所有顶点都得到了标号(见图3.4，得到标号的次序是 v_1, v_4, v_6, v_3, v_2, v_7, v_5)，从图3.4可以看出，从 v_1 到 v_7 的最短有向路的长度是13，并且最短有向路是：

$$p_1=\{v_1, v_4, v_6, v_7\},$$

从 v_1 到其他顶点的最短有向路也都找出来了，例如从 v_1 到 v_2 的最短有向路的长度是7，最短有向路是

$$p_2=\{v_1, v_4, v_6, v_2\},$$

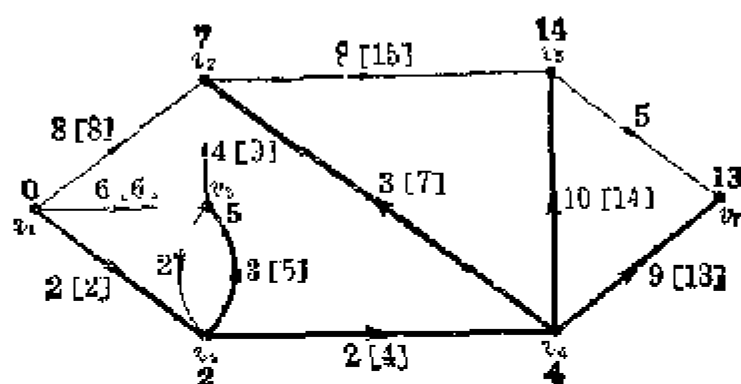


图 3.4

在这个例子中，如果我们只要求 v_1 到 v_7 的最短路，那末也可以在 v_7 得到标号后，就结束计算，从而省去了最后一轮计算.

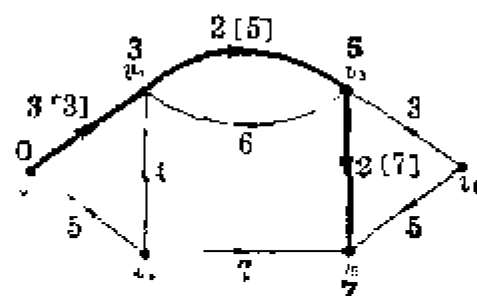


图 3.5

如果在计算结束时，还有一些点没有得到标号，那末，可以肯定，从 v_1 到这些点的有向路是不存在的（下一节还要证明这一点）. 例如，对于图3.5中的图，已经进行了三轮计算. v_1, v_2, v_3, v_5 四个顶点得到了标号，在进行第四轮计算时，因为不存在起点为已标号点而终点为未标号点的弧，故应结束计算. 但顶点 v_4 与 v_6 还没有得到标号，因此，按照前面讲的可以肯定，不存在从 v_1 到 v_4 和 v_1 到 v_6 的有向路，读者可以看出，图中 v_1 到 v_4 确实没有有向路.

图3.6中画的是上面讲的计算方法的框图.

在讲这个方法的证明以前，请大家先做几个练习. 把这个方法掌握得比较熟练以后，再学习证明就容易得多了.

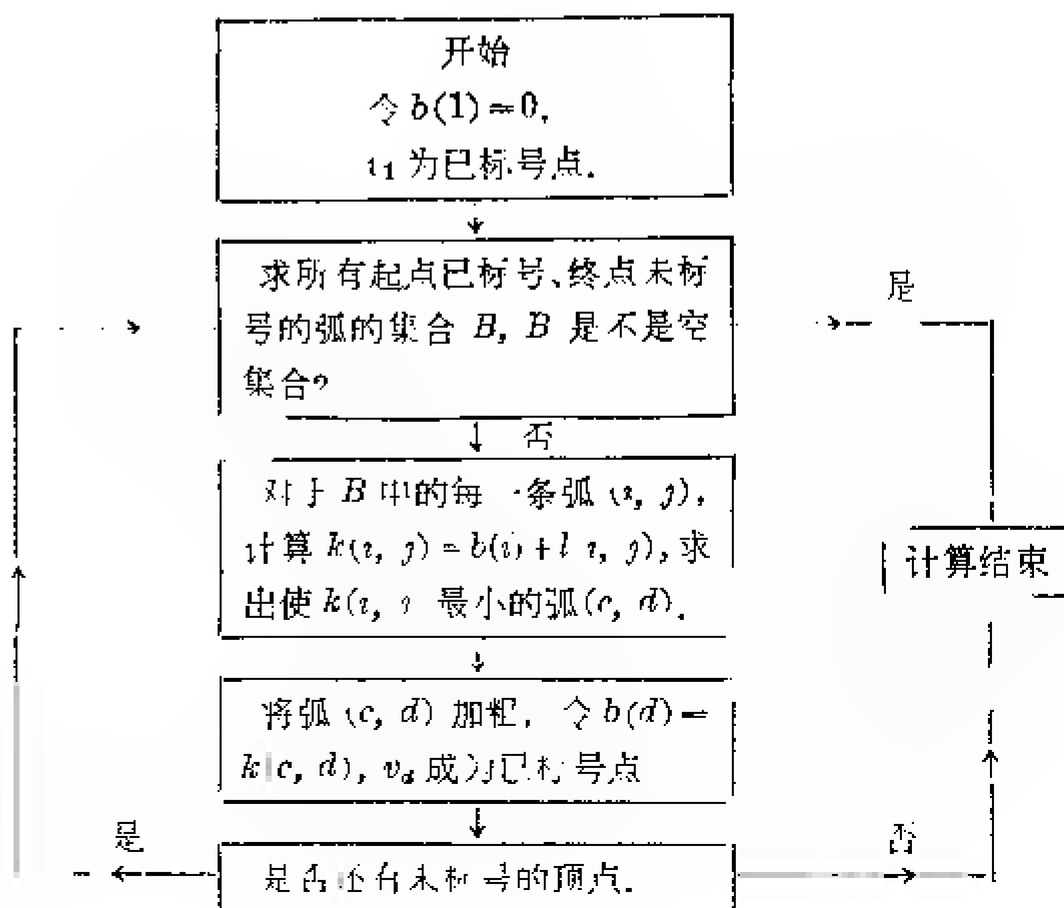
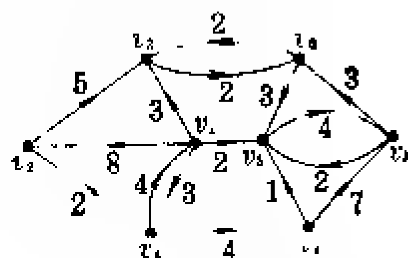


图 3.6

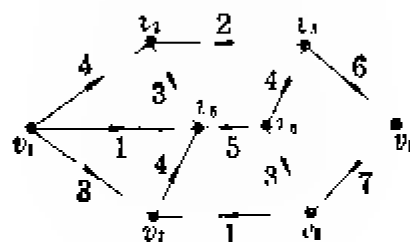
习 题

1. 求下图中从 v_1 到各个顶点的最短有向路，如果只要求出 v_1 到 v_8 的最短有向路，可以少进行几轮计算?

2 求下图中从 v_1 到其他顶点的最短有向路，指出对于 v_1 来说，哪些顶点是不可达的.



(第 1 题)



(第 2 题)

3. 标号法的证明

现在就来证明上节讲的标号法是正确的。应该注意，证明一个方法是"正确的"，和几何中证明一个定理成立有点不一样。几何中的定理，需要证明的结论是什么，写得清清楚楚。但是为了证明一个方法是正确的，应该证明些什么呢？往往不很明确。因此必须先进行一些分析，看看证明了哪些结论以后，才可以大胆地说"这个方法是正确的"。

回想一下上一节讲的方法，不难看出，我们应该证明下面两个结论：

1. 设在某一轮计算中，顶点 v_j 得到了标号 $b(j)$，那末：

(1) 存在从 v_1 到 v_j 的由粗线组成的有向路；

(2) 从 v_1 到 v_j 的由粗线组成的有向路的长等于 $b(j)$；

(3) 图 G 中所有其他的从 v_1 到 v_j 的有向路的长大于或等于 $b(j)$。

不难看出，证明了上面的 (1)(2)(3) 以后，就可以肯定 $b(j)$ 是 v_1 到 v_j 的最短有向路的长，而由粗线组成的路一定是最短有向路了。

2. 如果在计算结束时，顶点 v_j 还没有得到标号，那末在图 G 中一定不存在从 v_1 到 v_j 的有向路。

请大家在看后面的证明以前，再结合着上一节的方法想一想，证明了上述几个结论是不是足以说明标号法是"正确的"了。

先证明第 1 个结论。为了使大家容易学，我们结合着图 3.2 的例子来证明，一般情况的证明是完全一样的。

我们将按照得到标号的次序，一个顶点一个顶点地证明结论1成立．例如在图3.4中，除了v_1以外，得到标号的次序是v_4，v_6，v_3，…．

先来看v_4，它是在第一轮计算中得到标号的．第一轮计算中考虑的弧实际上就是所有从v_1出发的弧，在图3.2中，就是弧(1, 2)、(1, 3)和(1, 4)．因为$b(1)=0$，所以$k(i, j)=l(i, j)$，因而使$k(i, j)$达到最小的弧也就是从v_1出发的弧中的最短的弧(1, 4)．在第一轮计算结束时，我们把弧(1, 4)加粗，并且给v_4以标号$b(4)=k(1, 4)=l(1, 4)$，下面就来证明，对于v_4来说，结论1中的(1)(2)(3)都成立．

(1)和(2)是显然的，因为单单一条粗弧(1, 4)就是v_1到v_4的由粗弧组成的有向路，它的长度恰好等于$b(4)$．

现在来证明(3)．设p是任意一条从v_1到v_4的有向路(见图3.7)，那末p的第一条弧一定是(1, 2)、(1, 3)、(1, 4)这三者之一．但是这第一条弧的长度已经大于等于(1, 4)的长度了，当然p的长度就更大于等于(1, 4)的长度了．

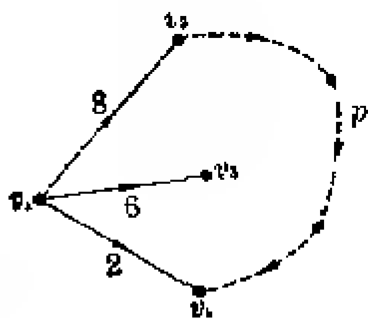


图 3.7

再来看看第二轮计算中得到标号的顶点v_6．这时，首先要记住，对于前面已经得到标号的顶点v_1、v_4来说，结论1中的(1)(2)(3)都已经证明是成立的了(对于v_1，结论1中的(2)是显然成立的，而(1)和(3)也不需要证明)．

在这轮计算中，使$k(i, j)$最小的弧是(4, 6)，因此我们把弧(4, 6)加粗，并且令$b(6)=k(4, 6)=2+2=4$．

先证明(1)和(2)成立．因为v_4是已标号点，对于v_4，(1)

(2)成立．所以存在从 v_1 到 v_4 的由粗弧组成的有向路，它的长度是 $b(4)$．在这条有向路上加上刚刚加粗的弧 $(4, 6)$，就得到一条从 v_1 到 v_6 的由粗弧组成的有向路，它的长度正好是：

$$b(4)+l(4, 6)=k(4, 6)=b(6).$$

再来证明 (3)．设 p 是任意一条从 v_1 到 v_6 的有向路，因为 v_1 是已标号点，而在这轮计算结束前，v_6 是未标号点，因此在 p 上一定有一条弧，它的起点为已标号点而终点为未标号点（只要从 v_1 开始沿着 p 的方向往前走，遇到第一个未标号点时，p 上的以这个未标号点为终点的弧，就是一条这样的弧）．也就是说 p 一定包含这轮计算的步骤 1 中找出来的 4 条弧：$(1, 2)$、$(1, 3)$、$(4, 3)$、$(4, 6)$ 中的一条，例如设 p 包含了 $(4, 3)$（见图 3.8，包含其他弧也一样），即：

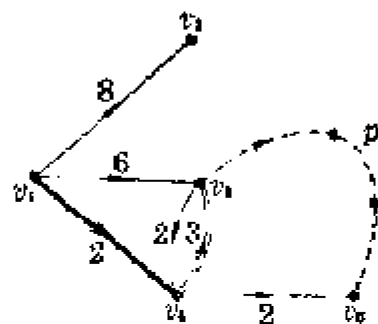


图 3.8

$$p=\{v_1, \cdots, v_4, v_3, \cdots, v_6\},$$

这时，因为 v_1 到 v_4 的最短有向路的长度是 $b(4)$，所以 p 中从 v_1 到 v_4 这一段的长度至少是 $b(4)$，而从 v_1 到 v_3 这一段的长度就至少是 $b(4)+l(4, 3)=k(4, 3)$ 了，但 $k(4, 3)\geqslant k(4, 6)$，因为 $k(4, 6)$ 是四个 $k(i, j)$ 中最小的．因此，

$$p \text{ 的长度} \geqslant p \text{ 中从 } v_1 \text{ 到 } v_3 \text{ 这一段的长度} \geqslant k(4, 3)\geqslant k(4, 6)=b(6).$$

对于在以后各轮计算中得到标号的顶点，可以用完全相似的办法证明结论 1 中的(1)(2)(3)都成立，这里就不仔细写了．

再来证明结论 2 也成立．即假设计算结束了，但是某一个顶点 v_j 仍没有得到标号，来证明一定不存在从 v_1 到 v_j 的

有向路. 回想一下，整个计算在什么情况下会结束？有两种可能：一是在某一轮计算的步骤 I 中找不到起点已标号而终点未标号的弧，另一是所有顶点都得到标号了. 因为已知 v_t 没有得到标号，所以出现的一定是前一种情况，但是这时就很容易证明不存在从 v_1 到 v_t 的有向路了. 因为如果存在从 v_1 到 v_t 的有向路 p，那末因为 v_1 已标号而 v_t 未标号，和前面讲的一样，在 p 上一定有起点已标号而终点未标号的弧，这就与刚才讲的找不到具有这种性质的弧矛盾了.

到此，标号法的正确性就证完了.

习　题

试试看，对于一般情况证明标号法的正确性。

4. 标号法好不好

现在来讨论，前面讲的标号法好不好？要回答这个问题，首先应该明确一下什么叫"好"，什么叫"不好". 一般说来，主要的好坏标准是计算起来快不快(还有别的标准，例如容易不容易拿上电子计算机计算；是否易于普及等等，这里就不仔细讲了)，或者说，用这个方法计算时，需要进行的运算次数多不多. 当然，运算次数愈少愈好.

大家也许会说，运算次数多少不完全决定于采用什么方法，还和要解决的问题有关. 同样用标号法，解一个只有 10 个顶点的问题可能只要进行几千次运算，而解一个 100 个顶点的问题，就可能要进行几百万次运算了，这又怎么比较呢？

办法还是有的. 那就是，设图 G 有 n 个顶点（为了简单

起见，我们就不研究边数 m 的影响了），我们来估计一下，把标号法用到图 G 上去需要进行几次运算．当然，这样估计出来的结果不会是一个确定的数，而是象 n^2, $3n^3+4m^2$, 2^n 等等这样的与 n 有关的数，即 n 的函数．然后再以这种函数为标准来比较方法的好坏．比如说，有两种方法，第一种要进行 n^3 次加法，而第二种要进行 n^5 次加法，当然第一种就比第二种好，因为在 n 较大时，n^5 比 n^3 要大多了．

上面讲的是怎样比较两种方法谁好谁坏．现在总共只讲了一个标号法，又怎么评论它的好坏呢？也有办法的．目前一般认为，如果一种方法所需要的运算次数能表示成 n 的多项式，例如 n^4, $2n^2+3n$ 等等．这种方法就认为是好的，或者说是有效的．而如果一种方法的计算次数是某一个数的 n 次幂，例如 2^n, 10^n，即是 n 的指数函数，这种方法就认为是不好的，或者说是无效的．为什么订这样的标准呢？请大家看看下面这张表：

表 3.1

n	5	10	20	30	50	100	1000
方法 A（运算次数 n^3）	625	1000	8000	27000	125000	10^6	10^9
方法 B（运算次数 2^n）	32	1024	$\approx10^6$	$\approx10^9$	$\approx10^{15}$	$\approx10^{30}$	$\approx10^{300}$

表中对一种需要进行 n^3 次运算的方法 A 与另一种需要进行 2^n 次运算的方法 B 进行了比较（关于 2^n 的近似值，我们是以 $2^{10}=1024\approx10^3$ 来估算的，例如 $2^{50}=(2^{10})^5\approx(10^3)^5=10^{15}$）．从表 3.1 可以看出，方法 B 的运算次数的增长速度是惊人的．

也许有的读者会认为，现在反正有大型电子计算机，计算次数多一些无所谓．其实不然．例如我们有一个每秒能计算一百万次的计算机，那末，在1000秒内可以进行 $1000\times1000000=10^9$ 次(即十亿次)运算，如果用方法A，则可以解决一个1000个顶点的问题，而用方法B呢？却只能解决一个30个顶点的问题．如果想用方法B来解决一个100个顶点的问题，即使用的是每秒能计算一亿次的计算机，也需要 10^{22} 秒，即要好几万亿年！

从上面的简单比较就可以看出，为什么说计算次数是 n 的多项式的方法是有效的，而计算次数是 n 的指数函数的方法是无效的．另外，也可以看出，单靠提高计算机的速度还不够，还必须从数学上寻求有效的计算方法．

现在再回过头来看看标号法好不好．回想一下标号法的各轮计算，可以看出，它只包含两种运算：加法与比较大小(比较大小也需要花费时间，所以也要考虑)．加法用于计算 $k(i, j)$，每计算一个 $k(i, j)$ 进行一次加法，而且每一条弧最多只计算一次．因此，如果图中有 m 条弧，那末至多进行 m 次加法．对于一个有 n 个顶点的简单有向图来说，最多有 $n(n-1)$ 条弧(假设从每一个顶点 v_i 出发，都有 $n-1$ 条弧指向其他的 $n-1$ 个顶点)，因此，最多进行 $n(n-1)$ 次加法，放宽一点，也可以说，至多进行 n^2 次加法．

另外，在每一轮计算中，在找使 $k(i, j)$ 达到最小的弧 (c, d) 时，要用到比较大小的运算，一般说来，要从 s 个数中把最小的数找出来，要进行 $s-1$ 次比较(例如有 a_1, a_2, a_3, a_4 四个数，那末可以先拿 a_1 与 a_2 比，然后拿这两个数中小的数与 a_3 比，再拿小的数与 a_4 比，比三次就能知道哪个数最小

了）。那末在每一轮的步骤1中，一般会选出几条弧呢？算得宽一些，至多 n^2 条吧（事实上要少得多）。因此至多进行 n^2 次比较，整个计算的轮数不会超过 n，因此，总起来说，至多进行 n^3 次比较大小的运算。

通过上面的估计，可以得出这样的结论：把标号法用在一个 n 个顶点的图上，至多进行 n^3 次加法和 n^3 次比较大小。因此，可以说，标号法是一种好的、有效的计算方法。

5. 求无向图上最短路的方法

首先，前面第2小节中讲的标号法也可以用到无向图上来。例如我们要找下面图3.9的无向图 G 中，连接顶点 v_1 与 v_8 的最短路。这时，我们就把图中的每一条边看成是两条方向相反的弧，即把图 G 变成图3.10中的 G'。然后在图3.10上用标号法求出从 v_1 到 v_8 的最短有向路，计算的结果已写在图3.10上了（请大家自己先算算，然后再和图3.10中的结果对照一下）。

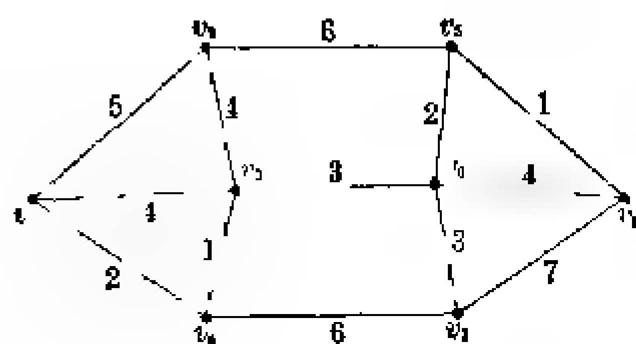


图 3.9 G

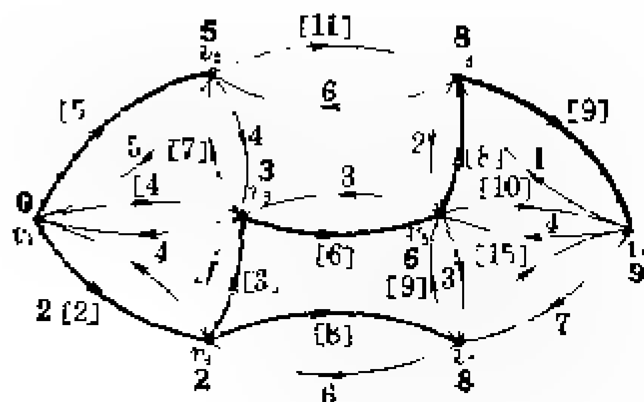


图 3.10 G'

我们在 G' 中找到

的最短有向路是：

$$p' = \{v_1, v_4, v_3, v_6, v_5, v_8\}$$

$$= \{(1, 4), (4, 3), (3, 6), (6, 5), (5, 8)\}$$

它的长度是 9. 这时，我们就可以肯定，图 3.9 的 G 中的路：

$$p = \{v_1, v_4, v_3, v_6, v_5, v_8\}$$

$$= \{[1, 4], [4, 3], [3, 6], [6, 5], [5, 8]\},$$

就是要找的连接 v_1 与 v_8 的最短路了.

为什么 p 一定是最短路？道理很简单. 如果 G 中还有比 p 更短的连接 v_1 与 v_8 的路，比如说，G 中的路

$$p_1 = \{v_1, v_2, v_5, v_8\}$$

比 p 短，这时，很显然，在 G' 中一定也会找到一条有向路：

$$p_1' = \{v_1, v_2, v_5, v_8\}$$

比 p' 短，这就与 p' 是 G' 的最短有向路矛盾了.

讲到这里，大家可能会感到：很好，求最短无向路的问题已解决了. 不过，还有一件事应该考虑，就是：如果为了求一个无向图上最短路，每次都要画象图 3.10 这样的一张图，似乎太麻烦了吧？能不能不画图 3.10，而直接就在图 3.9 上求呢？答案是肯定的. 事实上，只要把第 2 小节中的每一轮计算的各个步骤中的弧都改为边，把步骤 1 中的"起点 v_i 是已标号点而终点 v_j 是未标号点"改为"一端是已标号点而另一端是未标号点"，就可以直接在无向图上求最短路了. 建议大家在图 3.9 上试试看，然后再把这个计算方法总结一下.

下面再讲一种求无向图上最短路的方法，叫做"结网模拟法". 方法是：设 $G=[V, E]$ 中有 m 条边 $e_1, e_2, \cdots, e_m$. 我们取 m 条绳子，与这 m 条边对应，并且使每根绳子的长度与对应边的长度成比例，然后把这 m 根绳子按照 $G=[V, E]$

的样子编成一个网．如果要求连接 v_3 与 v_6 的最短路，只要用两手分别拿住网中代表顶点 v_3 与 v_6 的地方，再把网拉紧，这时拉成直线的绳子就对应着 G 中连接 v_3 与 v_6 的最短路了．

这种方法的道理是很显然的．对于不太复杂的问题，计算起来也是比较简单的．

习　　题

1　总结一下在没有负权的有向图上求最短路的标号法，并画出计算框图．

2.　在图3.9上求连接 v_2 与所有其他顶点的最短路（即连接 v_2 与 v_1，v_2 与 v_3，…，v_2 与 v_8 的最短路）．

3　求4节的图0.1中连接 v_1 与 v_{10} 的最短路．

4.　用这一节学的方法，为本章第1小节的渡河问题找一个解．

6．图的距离表

在生产实际中，往往需要求出一个图的任意两个顶点之间的最短路．例如，一个铁路局在编制他所属的各个车站之间的运输里程表时就会遇到这样的问题．另外，对于不少图论中的极值问题（例如下一章将要讲的服务点设置问题），往往在计算前首先必须把图中任意两个顶点之间的最短路及最短路的长度都求出来．

要把一个图的任意两个顶点之间的最短路都求出来，并不困难．例如，在有向图上，用第2小节的方法，计算一次就可以把从 v_1 到所有其他顶点的有向路都求出来了．接着从 v_2 出发，就是一开始令 v_2 为已标号点，并且令 v_2 的标号 $b(2)=0$，又可以把从 v_2 到其他各个点的最短路都求出来，……．这

样运算 n 次，就可以把从任意一个顶点到另一个顶点的最短路都求出来了. 对于无向图来说，也可以用相似的方法来解决.

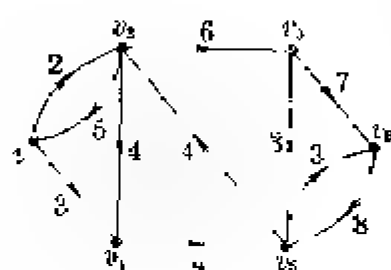
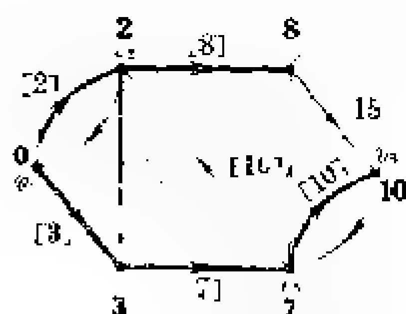


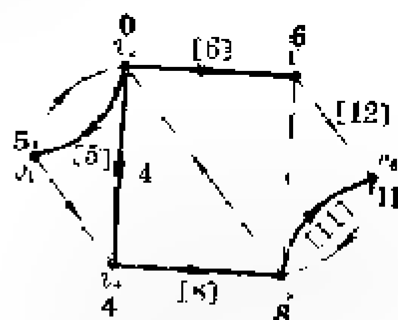
图 3.11

以图 3.11 中的有向图为例，这个图有 6 个顶点，在图 3.12 的 (a) 到 (f) 中分别画了以 $v_1, v_2, \cdots, v_6$ 为起点的计算结果.

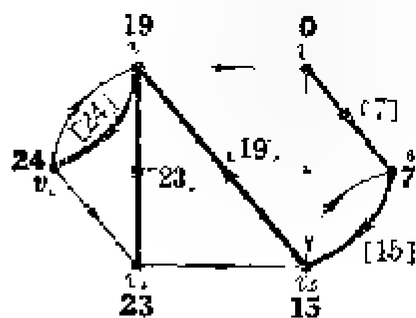
例如从图 3.12(c) 可以看出 v_3 到 v_1 的最短路的长是



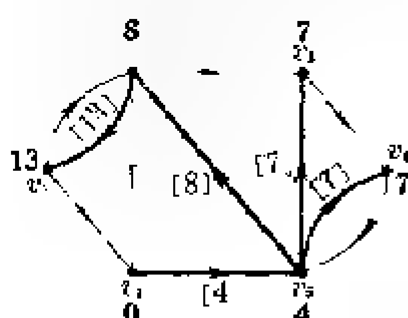
(a) 起点 v_1



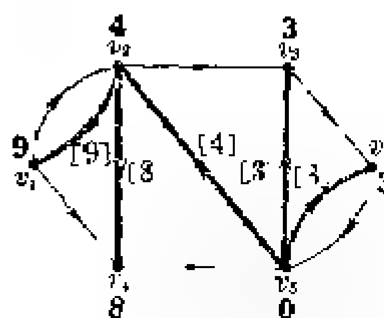
(b) 起点 v_2



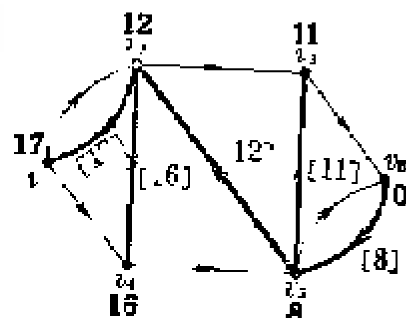
(c) 起点 v_3



(d) 起点 v_4



(e) 起点 v_5



(f) 起点 v_6

图 3.12

24，最短路是：

$$p_1=\{v_3,\ v_6,\ v_5,\ v_2,\ v_1\},$$

又如从图3.12(f)可以看出，从 v_6 到 v_2 的最短路的长是12，最短路是：

$$p_2=\{v_6,\ v_5,\ v_2\}.$$

为方便起见，今后我们把从顶点 v_i 到 v_j 的最短路的长叫做 v_i 到 v_j 的距离，记作 $d(v_i,\ v_j)$ 或 $d(i,\ j)$。

从上面的六张图虽然可以查出任意一个顶点到另一个顶点的距离，但是这样做毕竟不太方便．比较方便的办法是把这些距离集中写在一张象表3.2那样的表上．在这张表上，横着排的一行数字叫做"行"，例如0，2，8，3，7，10这6个数字组成表的第一行，9，4，3，8，0，3组成第5行；竖着排的一行数字叫做"列"，例如8，6，0，7，3，11这6个数字就是表的第3列．在表的第1行上，写的是从 v_1 到各个顶点的距离，也就是图3.12(a)中各个顶点的标号．同样第2行是 v_2 到各顶点的距离，……．而第1列呢，则是从各个顶点到 v_1 的距离，其余各列也一样．

象表3.2这样的表以后叫做图 G 的距离表．从表上查

表 3.2

	v_1	v_2	v_3	v_4	v_5	v_6
v_1	0	2	8	3	7	10
v_2	5	0	6	4	8	11
v_3	24	19	0	23	15	7
v_4	13	8	7	0	4	7
v_5	9	4	3	8	0	3
v_6	17	12	11	16	8	0

从顶点 v_i 到顶点 v_j 的距离 $d(i, j)$ 要比从前页的六张图上查容易得多了．例如 $d(3, 1)$ 就是表中第 3 行第 1 列交叉处的数，等于 24；又如 $d(5, 6)=3$ 等等．当然这张表也有不够的地方，例如不能查出从 v_i 到 v_j 的最短路是由哪些弧组成的．

对于无向图，我们仍用 $d(v_i, v_j)$ 或 $d(i, j)$ 表示连接顶点 v_i 与 v_j 的最短路的长度，并称它为 v_i 与 v_j 间的距离．当然，对于无向图，也可以用类似的办法造出距离表．

前面已经讲过，用标号法求最短路是有效的，因此，连用 n 次标号法来求距离表也是有效的．

习 题

1. 求本章第 2 小节习题中的图的距离表．
2. 求本章第 5 小节图 3.9 中的无向图的距离表．

四、服务点设置问题

1. 第一种标准，使最大服务距离达到最小

服务点设置问题，也是一个应用很广的图论中的极值问题。在引言中讲的公社卫生院地点的选择问题，就是服务点设置问题的一个例子。在考虑商店、电影院、消防站等设在哪里较好时，也会遇到类似的问题。

服务点设置问题的一般提法是：设 $G=[V, E]$ 是一个连通的无向图，在它的各个顶点 v_i 上，有一些服务对象（例如病人）现在要设置一个服务点（例如卫生院），问服务点设在 G 的哪一个顶点上最好？

在引言中讲卫生院设置问题时曾讲过，在研究这类问题时，首先要解决的是好坏标准问题。在那里也已经讲过一个标准，就是以来看病的病人中，最远的病人走的路程为标准。或者说，要使最远的服务对象与服务点的距离尽可能小。这一节就讨论在这种标准下，如何求最好服务点的问题。

设 $G=[V, E]$ 是一个连通的无向图，每一条边 e_j 有一个非负的长度 $l(e_j)$。现在任取一个顶点 v_i，然后考虑 v_i 与 G 的所有顶点之间的距离：

$$d(v_i, v_1), d(v_i, v_2), \cdots, d(v_i, v_n),$$

我们就把这 n 个距离中的最大数称为顶点 v_i 的最大服务距

离，记作$e(v_i)$.

看一个例子. 图4.1的图G中，有7个顶点. 用本书第55页讲的办法，可以求出图G的距离表(见表4.1). 从距离表中，可以很容易地把每个顶点v_i对应的$e(v_i)$求出来. 例如v_1到各个顶点的距离正好是表4.1的第一行中的数，它们的最大值是9.3，所以$e(v_1)=9.3$，同理可以求出$e(v_2)=6.3$, $e(v_3)=5$, …, $e(v_7)=6.3$. 这些$e(v_i)$都写在表4.1的最右边了.

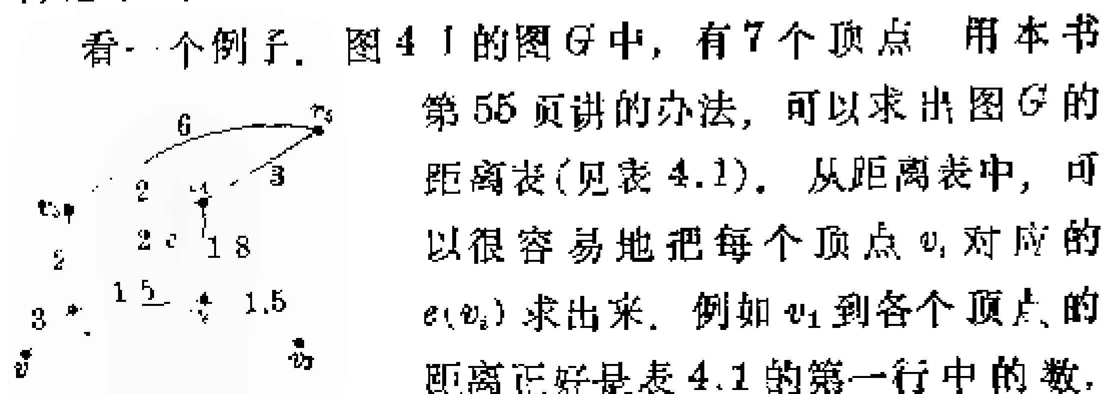


图 4.1G

表 4.1

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	$e(v_i)$
v_1	0	[illegible]	[illegible]	6.3	**9.3**	[illegible]	6	9.3
v_2	3	0	2	3.3	**6.3**	1.5	3	6.3
v_3	**5**	2	0	2	**5**	[illegible]	4	5
v_4	**6.3**	3.3	2	0	3	1.8	3.3	6.3
v_5	**9.3**	6.3	5	3	0	4.8	6.3	9.3
v_6	4.5	1.5	2.5	1.8	**4.8**	0	1.5	4.8
v_7	6	3	4	3.3	**6.3**	1.5	0	6.3

$e(v_i)$的实际意义是很清楚的. 就是：如果把服务点设在v_i，那末服务点与最远的服务对象间的距离就是$e(v_i)$. 我们给$e(v_i)$取"最大服务距离"这样一个名称的理由也就在这里.

有了$e(v_i)$立刻就可以判断服务点设在哪里最好了. 例如要在图4.1中选一个服务点，那末从表4.1可以看出，设在v_3最好，因为$e(v_6)=4.8$是所有$e(v_i)$中最小的.

定义4.1 使 $e(v_i)$ 达到最小的顶点叫做图 G 的中心。

从前面讲的可以看出，图4.1中的 G 的中心是 v_1。

把前面讲的总结一下，可以得出下述结论。如果以最大服务距离的大小作为好坏标准，那末，服务点以设在中心为最好。

那末怎样求一个图的中心呢？这个问题其实前面都已经讲了。下面再简单总结一下。

求图 G 的中心的步骤是：

步骤1 用上一章讲的计算方法求出图 G 的距离表。

步骤2 求出每一个顶点 v_i 的最大服务距离 $e(v_i)$，$e(v_i)$ 等于距离表上与 v_i 对应的行中的最大数。

步骤3 求出使 $e(v_i)$ 达到最小的顶点 v_k，v_k 就是图 G 的中心。

从上述三个步骤可以看出，步骤2和3都是很简单的，只要大约进行 n^2 次比较大小就可以完成了。因此，大部分工作集中在计算距离表上。在上一章最后已讲过，用标号法来求距离表是有效的。因此，用上面讲的三个步骤来求中心也是有效的。

习　　题

求下面图 G 的中心。



2 第二个标准，使运输量达到最小

再考虑一个服务点设置问题的例子.

我们再把图 4.1 看成一个矿区(图 4.2，它就是图 4.1)，它有 7 个矿，分别在 $v_1, v_2, \cdots, v_7$ 处. 这 7 个矿每天的矿石产量分别是: v_1: 3 千吨，v_2: 2 千吨，v_3: 7 千吨，v_4: 1 千吨，v_5: 5 千吨，v_6: 1 千吨，v_7: 4 千吨.

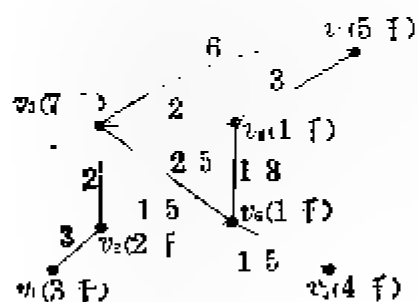


图 4.2

现在要从 $v_1, v_2, \cdots, v_7$ 中选一个地点来造选矿厂，问应该选哪一个地方，才能使各矿生产的矿石往选矿厂运时花费的运输力量最小?

在这个问题中，判断一个顶点好坏的标准就不能和前面讲的一样了. 对于一个顶点 v_i，这里我们应该计算的是，如果选矿厂设在 v_i，那末，将各个矿生产的矿石都运到 v_i 的运输量是多少? 然后再来比较在哪里设选矿厂最好. 一般，我们以运输的吨公里数来计算运输量的大小. 一吨货物运输一公里就叫一个吨公里，20 吨货运输 15 公里，就是 300 吨公里. 因此，拿上面的例子来说，如果选矿厂设在 v_1，则运输矿石共需:

$$g(v_1)=3000\times 0+2000\times 3+7000\times 5+1000\times 6.3+5000\times 9.3+1000\times 4.5+4000\times 6=122300\text{(吨公里)},$$

同样可以算出

$$g(v_2)=3000\times3+2000\times0+7000\times2+1000\times3.3+5000\times6.3+1000\times1.5+3000\times3=68300 \text{ 吨公里},$$

一比较，就可以看出选矿厂设在 v_2 要比设在 v_1 好. 当然，v_2 是不是最好的，还不能肯定，应该把 $g(v_3)$，…，$g(v_7)$ 都算出来，再比较一下，才可以选出一个最好的设厂点来.

仍拿上面的例子来说，我们可以在象表 4.2 这样的一张表上进行计算. 在表上，把各个顶点的产量放在最底下的一行上，计算出来的 $g(v_i)$ 写在最右边，比较一下就可以知道选矿厂以设在 v_3 为最好.

表 4.2

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	$g(v_i)$ (千吨公里)
v_1	0	3	5	6.3	9.3	4.5	6	122.3
v_2	3	0	2	3.3	6.3	1.5	3	68.3
v_3	5	2	0	2	5	2.5	4	64.5
v_4	6.3	3.3	2	0	3	1.8	3.3	63.5
v_5	9.3	6.3	5	3	0	4.8	6.3	108.5
v_6	4.5	1.5	2.5	1.8	4.8	0	1.5	65.8
v_7	6	3	4	3.3	6.3	1.5	0	88.3
产量 (千吨)	3	2	7	1		1	4	

上面我们虽然只是通过一个例子来讲，不过一般情况也是完全一样的. 一般地，设有一个连通图 $G=[V, E]$，G 的每一条边 e_j 有非负的长度 $l(e_j)$，G 的每一个顶点 v_i 还有一个

"产量" $a(v_i)$. 对于每一个顶点 v_i 令

$$g(v_i)=a(v_1)\cdot d(v_i, v_1)+a(v_2)\cdot d(v_i, v_2)$$
$$+\cdots+a(v_n)\cdot d(v_i, v_n).$$

$g(v_i)$ 代表把各个点的物资运到 v_i 所花费的吨公里. 然后,我们给出下面的定义:

定义 4.2 使 $g(v_i)$ 最小的顶点叫做图 G 的中央点.

按照前面讲的可以看出,在以运输量的大小为标准时,中央点就是设置服务点的最好位置了.

至于中央点的求法,只要看表 4.2 就明白了. 另外,不难看出,求中央点的计算方法也是有效的.

习　　题

1. 总结求图 G 的中央点的计算方法.

2. 本章第1小节的习题1中,设各个顶点的产量是

$a(v_1)=3$, $a(v_2)=5$, $a(v_3)=7$, $a(v_4)=2$,

$a(v_5)=5$, $a(v_6)=4$.

试求图的中央点.

3. p-中心问题

仍旧考虑第1小节中讲的公社卫生院的地点选择问题,不过现在不是设置一个卫生院,而是设置两个,问设置在什么地方最好?

和前面一样,首先还是要把好坏标准明确一下,仍旧以图 4.1 中的图 G 为例. 表 4.3 就是它的距离表(就是表 4.1). 现在我们先假设卫生院设在 v_2 和 v_5 两个地方. 这时,对于

表 4.3

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	0	3	5	6.3	9.3	4.5	6
v_2	3	0	2	3.3	6.3	1.5	3
v_3	5	2	0	2	5	2.5	4
v_4	6.3	3.3	2	0	3	1.8	3.3
v_5	9.3	6.3	5	3	0	4.8	6.3
v_6	4.5	1.5	4.5	1.8	4.8	0	1.5
v_7	6	3	4	3.3	6.3	1.5	0

表 4.4

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_2	3	0	2	3.3	6.3	1.5	3
v_5	9.3	6.3	5	3	0	4.8	6.3
[illegible]	3	0	2	3	0	1.5	3

住在 v_1 的居民来说，当他们要看病时，当然到离 v_1 较近的 v_2 去看病，因此 v_1 的居民因看病必须走的路程是3公里，同样的道理，可以求出 v_3, …, v_7 的居民看病时必须走的路程(见表4.4)，例如从表4.4中可以看出，住在 v_3, v_6, v_7 的居民一般也将到较近的 v_2 去看病，他们走的路程分别是2、1.5和3公里；而住在 v_4 的居民将到 v_5 去看病，要走3公里。从表4.4最下面的一行中还可以看出，当卫生院设在 v_2 和 v_5 时，住在各地的居民看病时需要走的路最多不超过3公里，我们

把这种情况记作 $e(v_2, v_5)=3$, $e(v_2, v_5)$ 叫做顶点对(就是一对顶点) v_2, v_5 的最大服务距离.

又例如从表 4.5 可以看出, 顶点对 v_1、v_7 的最大服务距离是 6.3, 即 $e(v_1, v_7)=6.3$. 和 $e(v_2, v_5)=3$ 一比较, 就可以看出设置在 v_2、v_5 比设置在 v_1、v_7 好.

表 4.5

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	0	3	5	6.3	9.3	4.5	6
[illegible]	6	3	4	3.[illegible]	6.3	1.5	0
距离	0	3	4	3.3	6.3	1.5	0

讲到这里,好坏标准问题就不难解决了,我们可以规定顶点对最大服务距离愈小, 就算愈好. 这样规定的实际意义和第 1 小节中是一样的,这里就不再仔细解释了.

和定义 4.1 相似,我们还可以给出下述定义.

定义 4.3 使 $e(v_i, v_j)$ 达到最小的顶点对叫做图 G 的 2 中心.

因此,要研究哪两个顶点设置卫生院最好,就归结为求图 G 的 2 中心了. 那末怎样求图 G 的 2 中心呢? 最简单的办法就是: 把 G 的所有顶点对 v_i、v_j 对应的最大服务距离 $e(v_i, v_j)$ 都找出来,比较一下,就可以求出 G 的 2 中心,从而也就知道在哪两个顶点上设卫生院最好了.

大家也许会感到:"这也算是一种方法吗? 真是笨方法!" 不过不要轻易下结论. 因为上一章曾经讲过评价一个方法好坏的办法,对于一个有 n 个顶点的图 G 来说,用这种"笨方法"

计算的话，在求出距离表（前面讲过，用标号法求距离表是有效的）后，还要计算 $\frac{n(n-1)}{2}$ 个顶点对的最大服务距离，而每次求最大服务距离只要进行约 $3n$ 次比较大小的运算，因此，总起来说，这种"笨"方法还是有效的.

2-中心这个概念还可以推广，即可以考虑3-中心、4-中心，…，或者一般地说可以考虑 p-中心. 在研究如何设置 p 个卫生院最好时，就会遇到求 p-中心的问题. 可以想象，p-中心问题的用途是很广的，但是，如果用与前面讲的笨方法相似的方法来求 p-中心，就不是一个有效方法了，因为这里计算步数依赖于顶点数 n 和 p，即计算次数是 n、p 的函数，而且可以证明这个函数不是 n 和 p 的多项式，而是指数函数. 因此 p-中心的算法问题还是一个未解决的难题.

习 题

求出图4.1的图 G 的2-中心.

五、最小树问题

1. 什么是最小树问题

这一章讲的最小树问题，是图论中又一个很重要的极值问题。它的重要性不亚于最短路问题。先讲一下什么叫树。

定义 5.1 设 $G=[V,E]$ 是一个无向图，如果它具有下述两个性质：

(1) 连通；

(2) 没有圈。

就称 G 是一个树(或一棵树)。

图 5.1(a)、(b) 中画的都是树的例子。也许有的读者会感到奇怪，为什么在数学中会采用"树"这样一个名词。不过，想象一下我们平常见到的树，就会发现如果把它们看成是由线和点组成的图形，它们确实都是连通而且没有圈的(见图

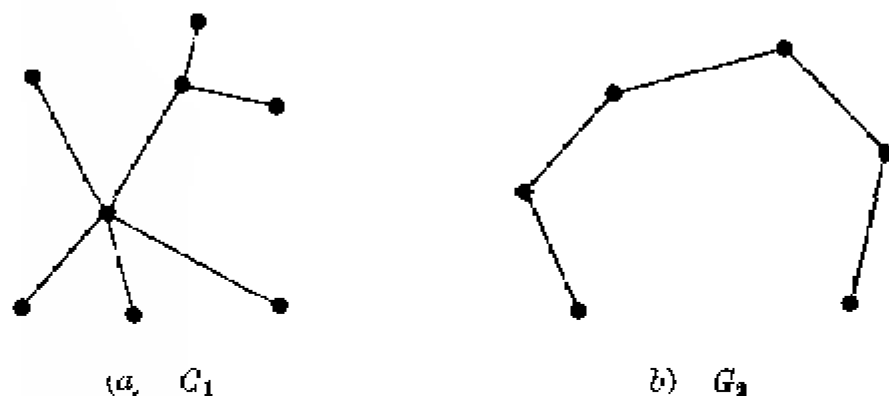


图 5.1



图 5.2 树

5.2)．有意思的是，许多图论中的概念在不同的书中有不同的名称，而树这个名称却是在所有的书中都一致采用的．

回想一下，在第二章中我们曾讨论过，所谓图 $G=[V, E]$ 的支撑子图，指的是 G 的一个子图 $G_1=[V_1, E_1]$，其中 $V_1=V$，即 G_1 是由 G 的全部顶点及一部分边组成的．对于我们来说，特别重要的是图 G（G 本身不一定是树）的那些形成树的支撑子图．

定义 5.2 设 $G=[V, E]$ 是一个无向图，如果 $T=[V, E_1]$ 是 G 的支撑子图并且 T 是树，则称 T 是 G 的一个支撑树．

例如图 5.3 (b) (c) 中的粗线及所有顶点组成的支撑子图，都是 (a) 中的图的支撑树．

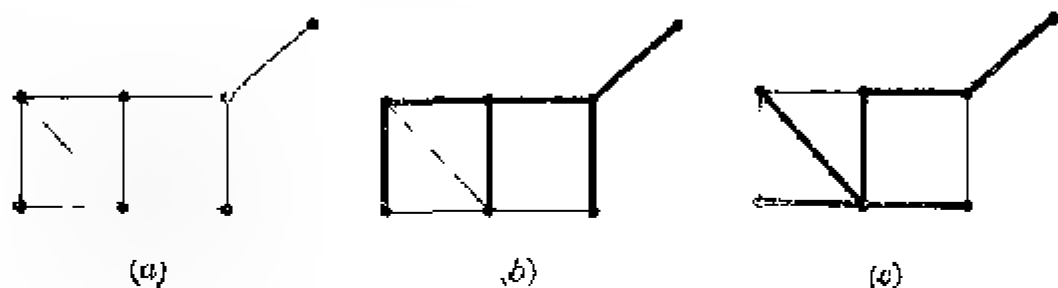


图 5.3

是不是每个图 G 都有支撑树呢？不见得．很显然，如果 G 不连通，G 就一定不会有支撑树．反过来，我们有：

定理 5.1 连通图一定有支撑树．

证明 设 G 是一个连通图，如果 G 没有圈，那末 G 本身就是一个支撑树，如果 G 有圈，那末任取 G 的一个圈，并且从这个圈中任意去掉一条边，得到 G 的一个支撑子图 G_1，易见 G_1 仍是连通的，如果 G_1 还有圈，就再从某一个圈中去掉一条边，得到 G_2，G_2 仍是连通的，……，这样做下去，直至得到一个不含圈的连通支撑子图 G_s 为止，G_s 就是 G 的一个支撑树了．证完．

图 5.4 中画的就是从一个连通图 G 出发，按定理 5.1 的证明方法得到一个支撑树的过程，这个过程可以称为"破圈"过程．

从刚才讲的破圈过程可以看出，一个连通图 G 一般有许多种支撑树．因为在取定一个圈后，可以从这个圈上任意去

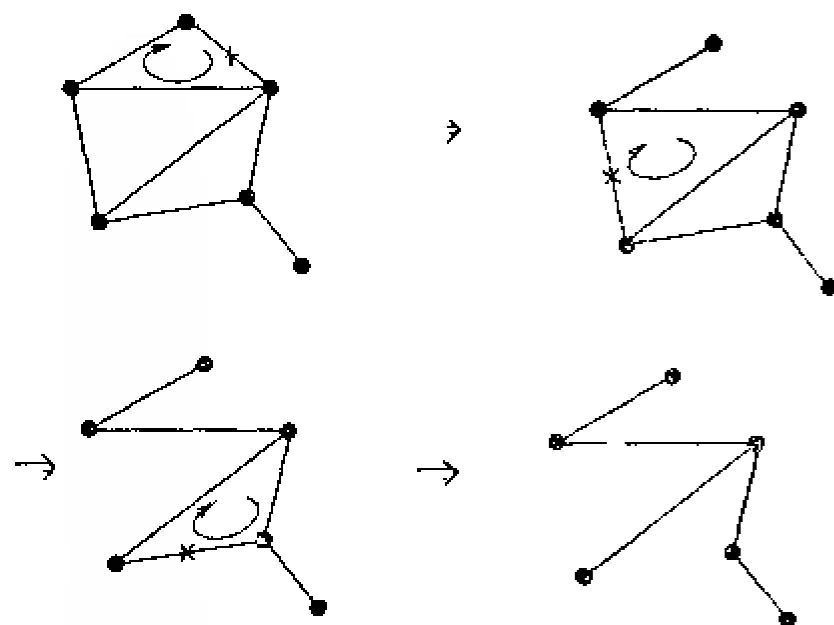
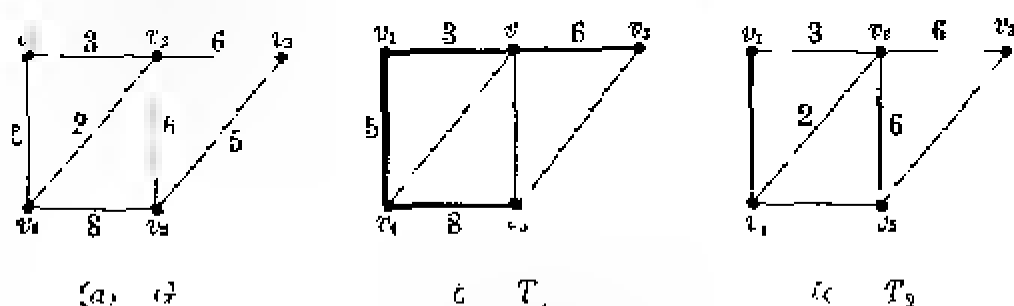


图 5.4

掉一条边. 去掉的边不一样,得到的支撑树就不同.

现在考虑一个连通图 $G=[V, E]$, 它的每一条边 e_i 有一个长度 $l(e_i)\geq 0$. 这时,对于 G 的任意一个支撑树 T, 我们把属于 T 的各条边的长度加起来的和叫做树 T 的长度,记作 $l(T)$. 例如对于图 5.5(a) 中的图 G 来说, (b)、(c) 中的两个支撑树 T_1 与 T_2 的长度就分别为 $l(T_1)=22$, $l(T_2)=17$.



(a) G　　(b) T_1　　(c) T_2

图 5.5

现在又可以提出一个极值问题了. 就是, 如何从 G 的所有支撑树中, 把长度最小的支撑树找出来?

通常, 我们把长度最小的支撑树叫做最小树. 有了这个名词以后,刚才提的极值问题就可以简单的说成是: 如何把最小树找出来. 因此,这个问题就叫做最小树问题.

最小树问题有很广泛的应用. 例如,我们把图 5.5(a) 的图 G 中的五个顶点看成某公社的 5 个大队, G 的边看成是公路,现在要架设电线(电线必须沿着公路架设),使各大队之间都能通电话,问应该怎样架线,才能使所用的电线最少?

考虑一下,就可以看出,这个问题的关键是决定图上哪些边上该架线, 哪些边上不架线. 设架线的边的集合是 E_1, 那末 $G_1=[V, E_1]$ 就是 G 的一个支撑子图. 因为架线后各个大队间都能通话,所以 G_1 必须是连通的. 因此要使电线最节约,就是要从 G 的所有连通的支撑子图中,把总边长最小的找

出来，但是不难看出，总边长最小的连通支撑子图一定不会含圈（想想看，为什么?），从而必定是一个支撑树．因此架设电线的问题就可以归结为最小树问题．

类似的问题还有许多，例如修公路把一些城镇连接起来，修渠道使水源和各块地连接起来，……，都可以归结为最小树问题．

另外，最小树问题还是许多其他图论中的极值问题的基础，也就是说，有不少问题在计算时，往往首先必须求出一个最小树．这也是最小树问题显得特别重要的一个原因．

在这一章的以后各节中，我们都假设遇到的图是简单图，这种做法的合理性与第二章中讲的是一样的（请看一下第二章§1最后一段），这里就不再细讲了．

习　　题

1．把图5.3(a)中的图G的所有支撑树都找出来，通过比较，求出G的最小树．

2．设图$G=[V, E]$的每条边e有长度$l(e)>0$，证明G的总边长为最小的连通支撑子图一定是支撑树．

2．树的等价定义

在介绍求最小树的计算方法以前，先讲一个定理，它在下一节讲计算方法时要用到．

定理5.2　设$T=[V, E]$是一个树，设T有m条边，n个顶点，则$m=n-1$．

在证明这个定理以前，先介绍一个概念并且证明两个引

理，这些概念和引理本身也是很重要的.

定义 5.3 设 $G=[V, E]$ 为一个无向图，v_i 是 G 的一个顶点. 我们把 G 中与 v_i 关联的边的条数叫做顶点 v_i 的次数，记作 $\deg v_i$.

例如图 5.6 中，

$$\deg v_1=2,\ \deg v_2=3,\ \deg v_4=4,$$

$$\deg v_3=\deg v_5=\deg v_6=1,\ \deg v_7=0.$$

一个图中的顶点可以按它的次数来分类.

定义 5.5 次数为奇数的顶点叫做奇顶点，次数为偶数的顶点叫做偶顶点. 特别地，次数为零的顶点叫做孤立点，次数为 1 的顶点叫做悬挂点.

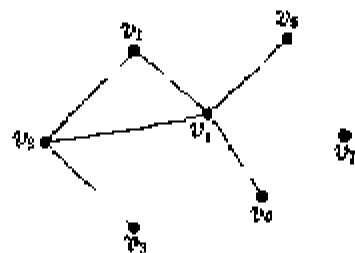


图 5.6

图 5.6 中，v_2, v_3, v_5, v_6 是奇顶点. 其余顶点都是偶顶点，v_7 是孤立点，v_3, v_5、v_6 是悬挂点. 从图 5.6 中可以看出，孤立点、悬挂点这些名词都是很形象的. 孤立点和任何别的点之间都没有路相连，确实是孤立的，悬挂点也确实象悬挂起来的点. 容易看出，一个连通图，只要顶点个数超过 1，就不会有孤立点.

引理 5.1 设 $G=[V, E]$ 是一个图，它的每一个顶点 v_i 的次数都满足 $\deg v_i\geqslant 2$，那末 G 一定有圈.

证明 证明的方法是：从任意一个顶点开始来构造 G 的一条简单路 p，开始时，p 只含一个顶点，以后逐步扩大，然后证明，扩大若干次后，p 中一定会出现圈，当然，这就证明了 G 中一定有圈了.

我们结合着图 5.7 中的图 G 来证明. 这个图的每一个

顶点的次数都 $\geqslant 2$.

先任意取一个顶点，例如取 v_4，并且令 $p=\{v_4\}$．因为 $\deg v_4\geqslant 2$，所以一定有与 v_4 关联的边，任取一条这样的边，例如取 e_4，把 e_4 及它的另一个端点 v_2 加到 p 中去，使 p 扩大成 $p=\{v_4, e_4, v_2\}$．然后再取一条与 v_2 关联，而不属于 p 的边．因为 $\deg v_2\geqslant 2$，这样的边是一定存在的．例如图 5.7 中可以取 e_1，把 e_1 及它的另一个端点 v_1 再加入 p，使 p 扩大成 $p=\{v_4, e_4, v_2, e_1, v_1\}$ ……，这样做下去，p 中每增加一条边 e'_i 与一个顶点 v'_i 后，就应该看一看，它属于下面两种情况中的哪一种．

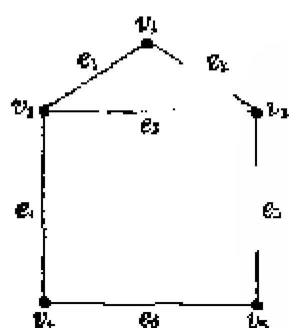


图 5.7

情况 1：v'_i 是第一次出现在 p 中．这时，因为 $\deg v'_i\geqslant 2$，所以一定还有与 v'_i 关联而不属于 p 的边，取一条这样的边，把它及它的不同于 v'_i 的另一个端点加入 p，p 就又可以扩大了．

情况 2：v'_i 是第二次出现在 p 中，这时不必再扩大 p 了．因为 p 中从上一次出现 v'_i 到这次出现 v'_i 中的一段就是一个圈．

因此，只要情况 2 一出现，就找到圈了．那么，情况 2 是不是一定会出现呢？一定会的．这是因为 p 是简单路，即每一条边在 p 中只出现一次，而图的总边数是有限的，因此，p 不能无限地扩大．要是在扩大 p 的过程中只是出现情况 1，那末 p 就可以不断地扩大下去．这个矛盾说明，经过若干次扩大后，一定会出现情况 2．

仍以图 5.7 为例，前面已扩大到 $p=\{v_4, e_4, v_2, e_1, v_1\}$

了．看一下 v_1，因为 v_1 是第一次出现在 p 中，属于情况1，故可以继续扩大．例如可以把 e_2 与 v_3 加到 p 中去．再看 v_3，仍是第一次出现．再扩大 p，例如取 e_3 与 v_2 即扩大成：

$$p=\{v_4,\ e_4,\ v_2,\ e_1,\ v_1,\ e_2,\ v_3,\ e_3,\ v_2\},$$

检查 v_2，v_2 是第二次出现，这属于情况2，故不必再扩大了，因为 p 中已出现了圈 $\{v_2,\ e_1,\ v_1,\ e_2,\ v_3,\ e_3,\ v_2\}$（见图5.8，带箭头的粗线表示 p 的扩大过程）．引理5.1证完．

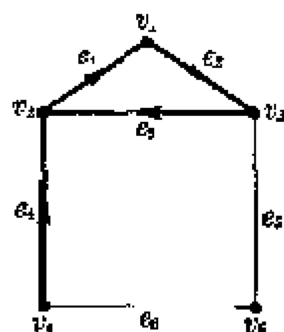


图 5.8

引理5.2 设 $T=[V, E]$ 为一个树，并且 T 至少有两个顶点，那么 T 一定有悬挂点．

证明 因为 T 是树，所以 T 是连通的，T 不会有孤立顶点，即每一个顶点 v_i 的次数 $\deg v_i>0$．如果 T 没有悬挂点，即 T 的每一个顶点的次数都大于等于2，那末，由引理5.1，T 含有圈，这就与 T 是树矛盾了．证完．

有了上面的准备，就可以来证明定理5.2了．

定理5.2的证明 设 $T=[V, E]$ 是树，如果 T 只有2个顶点，定理显然成立，现设 T 有不止2个顶点．由引理5.2，T 有悬挂点．设 v_i 是一个悬挂点．根据悬挂点的定义，应只有一条边与 v_i 关联，设这条边是 e_i，不难看出，由于 T 中有不止2个顶点，从 T 中去掉 v_i 与 e_i，剩下的仍是一个树 T_1（见图5.9）．因为 T_1 的边数和顶点数比 G 的边数和顶点数都小1．所以只要能够证明 T_1 的边数比顶点数少1，也就证明了 T 的边数比顶点数少1，从而也就证明了定理5.2了．

同样，因为 T_1 是树，T_1 也一定有悬挂点．如果 T_1 的顶点数 >2，则再去掉一个悬挂点和与它关联的唯一的边可以

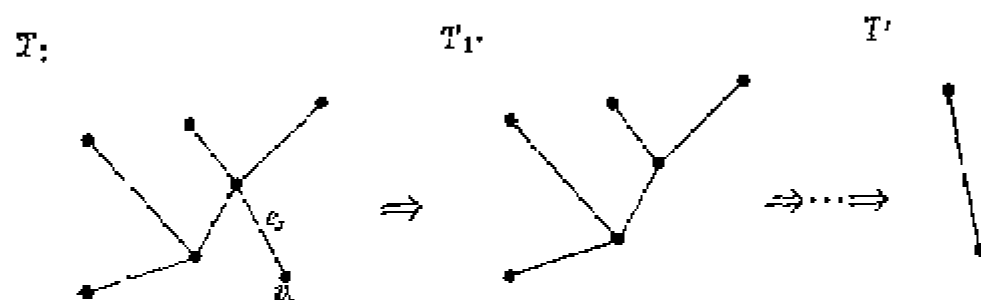


图 5.9

得到树 T_2，而且只要能证明 T_2 的边数比顶点数少 1，就能证明 T_1，从而 T 的边数比顶点数少 1 了，…… 这样不断的去掉悬挂点和边，一直到得到一个只含两个顶点的树 T' 为止．显然，T' 恰好含一条边．因此 T' 的边数比顶点数少 1，倒退回去，就可以证明 T 的边数也比顶点数少 1 了．证完．

重要的是，我们还可以证明

定理 5.3 设图 G 是连通的，并且边数等于顶点数减 1，那么 G 是树．

定理 5.4 设图 G 没有圈，并且边数等于顶点数减 1，则 G 是树．

这两个定理证明起来不很难，就留给大家作为习题吧．

上面三个定理合在一起，可以简单的说成：对于一个图来说，下面三个性质只要有两个成立，第三个也一定成立．

(1) 连通

(2) 没有圈．

(3) 边数等于顶点数减 1．

在定义 5.1 中，我们把具有上述性质 (1)、(2) 的图叫做树，在证明了前面的几个定理以后，我们就也可以把树定义为具有性质 (1)、(3) 或具有性质 (2)、(3) 的图了．也就是说，树这个概念可以有三种不同的方法来下定义．这三种定义当

然本质上是一样的，在数学中，一般称它们为等价的定义。

上面讲的树的等价定义，在求最小树时是很有用的。例如下一节讲的一个计算方法，求出的是一个满足(2)、(3)的支撑子图，由于树的等价定义，就可以肯定它是一个支撑树了。

习　题

1. 证明定理5.3与5.4.

2. 证明：如果图 $G=[V, E]$ 是一个树，则下述性质成立：对于 V 中任意两个不同的顶点 v_i 与 v_j，G 中存在唯一的一条连接 v_i 与 v_j 的初级路。反过来，如果一个图 G 具有上述性质，那末它一定是树。

3. 求最小树的两种计算方法

求连通图的最小树的方法很多，我们只讲其中的两种。第一种叫做破圈法。这个方法可以用一句口诀来概括，就是：

任取一个圈，去掉圈上最长的边。

让我们通过一个例子把这句口诀的用法解释一下。就拿图5.5(a)中的连通图 G 为例。G 有好几个圈，现在任取一个，例如就取：

$$p_1=\{v_1,\ v_2,\ v_4,\ v_1\}$$

这个圈上有三条边，最长的是 $[v_1, v_4]$，长度是5. 我们就把这条边去掉，从而也就把圈 p_1 "破"掉了(见图5.10)。接下去，再看看剩下的图中还有没有圈。如果没有，那末计算就结束了；如果有，就再任意取一个圈，再去掉圈上的最长边，把这个圈破掉，…，直到剩下的图上没有圈(或图上的边数等于顶点数减1)时为止。

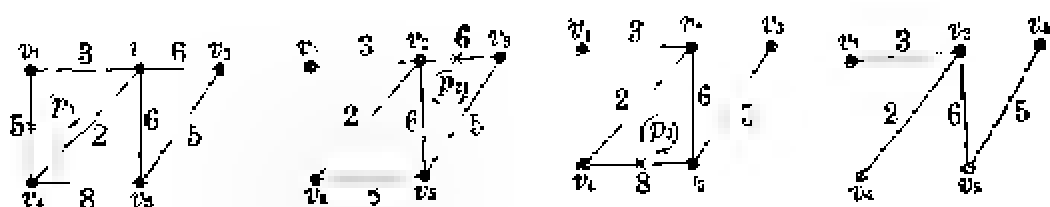


图 5.10 G

图5.10中画了整个破圈的过程，一共破三次，就得到了最小树．注意，在第二次取圈 p_2 时，圈上有两条边的长为6，都是最长边．这时，可以任意去掉其中一条．

做过第1小节习题1的读者，可以把做出的结果和图5.10比较一下，是不是一样？

破圈法讲起来很简单，但是，为什么这样做的结果得到的一定是最小树呢？证明起来倒有些麻烦，让我们在下面的第5小节中再讲吧．

再介绍一种求最小树的方法，叫做加边法．它与破圈法的做法正相反．破圈法是从原来的图中逐步去掉边，每次去的时候，要保持住图的连通性（从圈上去掉边，余下的图一定仍旧是连通的），直到图中没有圈为止．加边法正相反，它一开始先把图中的边都去掉，只留下孤立的顶点．然后逐步的把边加上去，每次加的时候，要保持住"没有圈"这一性质，在加了 $n-1$ 条边（n 是顶点个数）后，就得到要求的最小树了．

加边法的计算步骤是：

步骤1　把图 G 的 m 条边按从短到长的次序重新编号，即把最短的边叫做 e_1，次短的叫 e_2，…，最长的叫 e_m．

仍以图5.5(a)中的 G 为例，它的边按从短到长编号后，

应该编成象图5.11那个样了. G中的边e_3与e_4的长度都是5 因此这两条边的编号倒过来也可以.

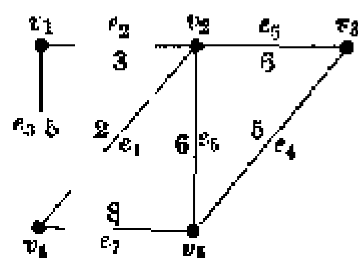


图 5.11 G

步骤2 首先把G的边都去掉,得到一个只含n个孤立顶点的支撑子图. 然后,按e_1, e_2, …, e_m的次序试着向支撑子图中加边. 对于每一条边e_j,要先看一看它是否和已经加进去的边形成圈. 如果不形成圈,就把e_i加进去,如果形成圈,e_j就不加进去,而考虑下一条边e_{j+1}. 一直加到得到的支撑子图含有$n-1$条边时(由定理5.4,这时,得到的支撑子图已经是支撑树了),计算结束.

图5.12中画的是用加边法求图5.11中的图G的最小树的过程.

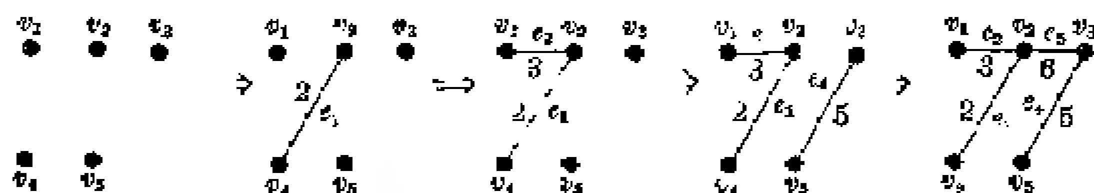


图 5.12

从图5.12可以看出,e_1, e_2加到支撑子图中去都不形成圈. 接着考虑e_3. e_3加进去恰好与e_1、e_2形成一个圈,因此e_3不能加. 而考虑下一条边e_4, e_4与e_5加进入都不形成圈,所以都可以加进去. e_5加进去后,得到了一个有4条边的支撑子图,边数已等于顶点数减1,因此应该结束计算.

图5.12中最后得到的最小树与图5.10的不一样,但是不难看出,它们的长是一样的,都是16.如果在对G的边编号

时，把 e_5 与 e_3 对换一下，得到的最小树就与图 5.10 一样了。

至于说为什么用这个方法求出的一定是最小树，也放在后面第 4 小节中证明。

破圈法与加边法哪一个好呢？应该说，两者各有优缺点。一般说来，图中圈较少时，用破圈法较好，圈很多时，用加边法较好。另外，破圈法讲起来简单，易于普及，而加边法呢，方法比较系统，易于编成程序在计算机上算。

图 5.13 中画的是加边法的计算框图：

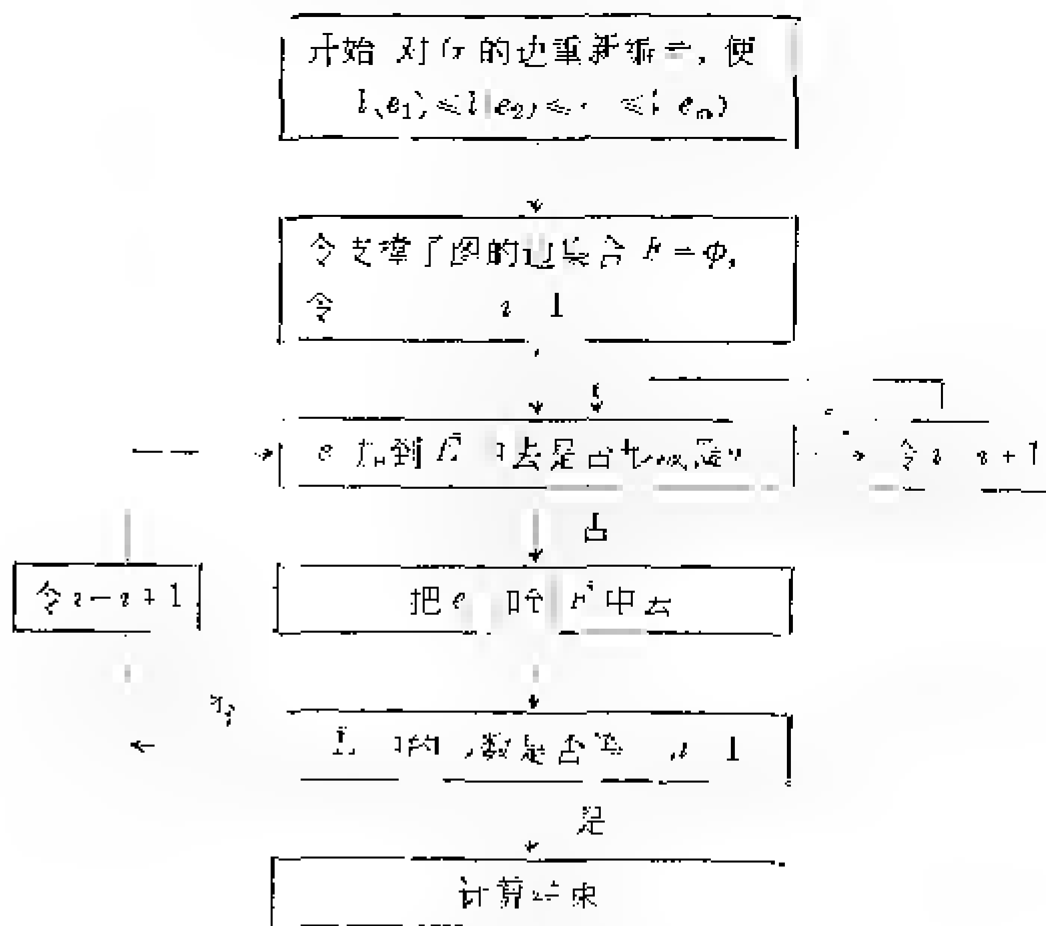


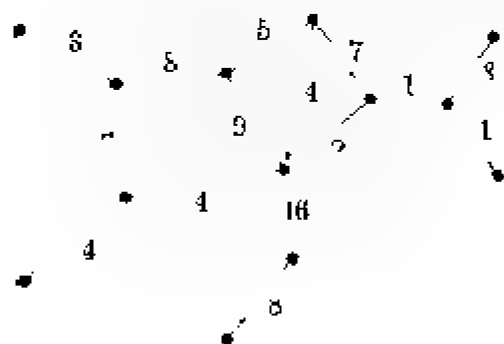
图 5.13

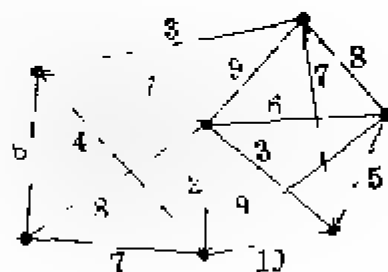
最后说明一下，以第三章第 4 小节讲的标准来衡量，加边法与破圈法都是有效的。不过，这一点就不详细讨论了。有兴趣的读者可以自己来估计一下用这两种方法来求最小树时

所需的运算次数.

习　题

1. 分别用破圈法与加边法求下面两个图的最小树.





4. 加边法的证明

这一节与下一节分别讲加边法与破圈法的证明. 这些证明读起来可能难一些,读者如果感到困难可以跳过去,这对下面几章的学习没有影响.

首先,再证明一个树的重要性质.

定理 5.5 设 $G=[V, E]$ 是一个连通图, T 是 G 的一个支撑树, e_i 是不属于 T 的一条边, 则在把 e_i 加入 T 后所得到的支撑子图 T' 中, 一定存在唯一的圈.

证明　设 $e_i=[v_s, v_t]$. 因为 T 是支撑树,是连通的,所以 v_s、v_t 属于 T, 并且在 T 中一定存在一条连接 v_s 与 v_t 的初级路. 不难看出 e_i 与这条初级路合在一起就是 T' 中的一个圈. (见图 5.14, 粗线代表属于 T



图 5.14

的边, 若把 e_8 加到 T 中去, T' 就有圈 $\{e_3, e_2, e_7, e_8\}$. 找找

看，把其他不属于T的边分别加到T中去，将得到哪些圈.）

再证明T'中只有唯一的一个圈. 用反证法，设T'中有两个不同的圈p_1与p_2. 显然，p_1与p_2一定都包含e_i（不然的话，T不加进e_i就有圈了，与T是树矛盾），而且除了e_i以外其余的边都属于T. 因此，p_1与p_2可以分别看成是e_i加上T的连接v_s与v_t的两条不同的初级路q_1和q_2组成，（见图5.15，双线为q_1，虚线为q_2），这时，$\underbrace{v_s, \cdots, v_t}_{q_1}, \underbrace{\cdots, v_s}_{q_2}$是一条回路.（图5.15中，从$v_s$出发，先沿着双线经$a$、$b$、$c$到$v_t$，再沿着虚线经$c$、$d$、$e$、$f$回到$v_s$），回路上的边都属于$T$. 另外，因为$q_1$与$q_2$不同，至少有一条边只出现在$q_1$或只出现在$q_2$中，也就是说，这条边在上述回路中只出现一次，根据第一章第4小节的习题3，不难看出，从这条回路中可以选出一个圈（图5.15中的最外面的圈$\{v_s, a, b, c, d, e, f, v_s\}$），圈上的边都属于$T$，这就与$T$是树矛盾了. 证完.

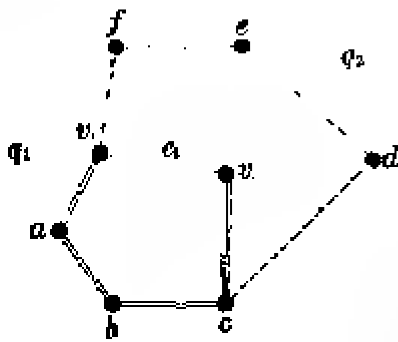


图 5.15

由定理5.5可以得出一种从一个支撑树变换成另一个支撑树的方法. 具体的做法是：

设T是图G的一个支撑树，e_i是不属于T的一条边，将e_i加到T中去，得到子图T'，由定理5.5可知T'中将有唯一的圈C. 然后在圈C上任意去掉一条边$e_j(e_j \neq e_i)$，把圈C破掉，就可以得到另一个支撑子图T_1了（见图5.16）.

上述把T变换成T_1的方法可以简称为“加边、破圈”变换.



图 5.16

有了上面的准备，就可以证明加边法的正确性了。首先我们作一个假定，就是：图 G 的各条边 e_i 的长都互相不相等。一般说来，对于任何生产实际中遇到的最小树问题来说，我们总可以认为这个假定是成立的。因为如果遇到上述假定不成立的图，例如图 5.5(a)，它的 e_3 与 e_4 的长都是 5，e_5 与 e_6 的长都是 6。这时我们就把 e_4 的长改为 5.00001，e_6 的长改为 6.00001，这样一改，上面的假定就成立了，而另一方面，很明显，这种修改不会产生实质性的影响。

作了上述假定以后，再将 G 的边按照从短到长的次序排列，就应该有：

$$l(e_1)<l(e_2)<\cdots<l(e_m).$$

定理 5.6 设用加边法求得的支撑子图为 T_1，则 T_1 是最小树。

证明 首先，用加边法求得的支撑子图 T_1 显然具有"没有圈"和"边数等于顶点数减 1"这两个性质，由第 2 小节讲的树的等价定义，可知 T_1 是支撑树。

现在来证明 T_1 是最小树。用反证法，设 T_1 不是最小树，而另一个支撑树 T 是最小树，然后设法来推出一个矛盾。

设 T_1 中的 $n-1$ 条边为 e_1', e_2', $\cdots$, e_{n-1}'，其中 e_1' 是第一个加到 T_1 中去的，e_2' 是第 2 个，$\cdots$，e_{n-1}' 最后加进去。由加边法的计算步骤可知：

$$l(e_1') < l(e_2') < \cdots < l(e_n').$$

T_1 中的 $n-1$ 条边中可能有一部分也属于 T，现在设 $e_1', e_2', \cdots, e_k'$ 都属于 T 而 e_{k+1}' 不属于 T，即按 $e_1', e_2', \cdots$，的次序往下看，e_{k+1}' 是 T_1 中第一条不属于 T 的边．将 e_{k+1}' 加到 T 中去，由定理 5.5，可以得到一个圈 C．不难看出，C 中至少有一条边 e_j，它不属于边集合 $E_1=\{e_1', e_2', \cdots, e_k', e_{k+1}'\}$．道理很简单，如果 C 中的边都属于 E_1，那末 C 就整个地属于 T_1 了（因为 E_1 中的边都是 T_1 的边），这与 T_1 是树相矛盾．

现在来证明这样的 e_j 一定满足 $l(e_j)>l(e_{k+1}')$．这是因为，首先 e_j 和 $e_1', e_2', \cdots, e_k'$ 都是 T 中的边，因此如果把 e_j 加到 $\{e_1', e_2', \cdots, e_k'\}$ 中去，不会形成圈．回想一下用加边法构造 T_1 的过程，在 $e_1', e_2', \cdots, e_k'$ 已经加入 T_1 以后，接着加进去的是 e_{k+1}' 而不是 e_j，这说明什么问题呢？说明 $l(e_{k+1}')<l(e_j)$，即在对 G 的边进行编号时，e_{k+1}' 排在 e_j 的前面（如果 e_j 排在前面，它又不与 $\{e_1', e_2', \cdots, e_k'\}$ 形成圈，它就应该先于 e_{k+1}' 加到 T_1 中去了），这就证明了 $l(e_j)>l(e_{k+1}')$ 了．

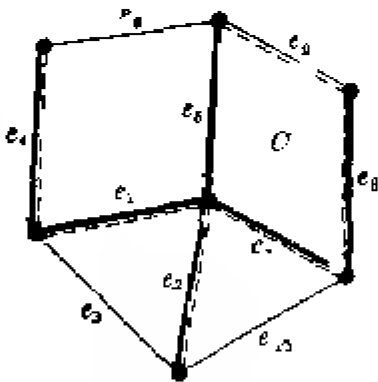


图 5.17

现在就可以得出一个矛盾了，办法就是：按照前面讲的"加边、破圈"变换，把 e_{k+1}' 加到 T 中去，然后去掉圈 C 上的边 e_j，得到一个新的支撑树 T_2．但是 $l(T_2)<l(T)$，因为从 T 变成 T_2 的过程中，增加的边 e_{k+1}' 比去掉的边 e_j 短．而这就与 T 是最小树矛盾了．定理 5.6 证完．

图 5.17 是定理 5.6 的证明过程的一张示意图．设图中的边已经从短到长编号，粗边表示由加边法求得的支撑树

T_1，虚线表示不同于T_1的最小树T. T_1中的边是：$e_1'=e_1$, $e_2'=e_2$, $e_3'=e_4$, $e_4'=e_5$, $e_5'=e_7$, $e_6'=e_8$，其中e_1', e_2', e_3'也属于T, $e_4'=e_5$是T_1中第一条不属于T的边. 将e_5加到T中去得到的圈C包含4条边$\{e_5, e_8, e_9, e_7\}$, e_7、e_8、e_9都可以选来作为e_j, 经过一次"加边、破圈"变换, 就可以得到一个长度比T小的支撑树了.

其实,仔细分析一下定理5.6的证明，可以看出，它实际上是证明了:"如果G的各条边的长都互相不同，那末任何与T_1不同的支撑树T都不是最小树". 因此我们还可以得到一个结论,就是:

定理5.7 如果G的各条边的长度互不相同，那末G的最小树是唯一的.

当然，在G有长度相等的边时，G的最小树不一定是唯一的。

习　题

证明在G中有长度相等的边时，用加边法求得的T_1仍是最小树。

5. 破圈法的证明

上一节证明加边法的正确性时,用了一种"加边、破圈"的变换方法. 为了证明破圈法的正确性，需要用另一种把一个支撑树变换成另一个支撑树的方法. 为了讲这种变换方法，先证明一个定理.

定理5.8 设$T=[V, E_1]$是图$G=[V, E]$的一个支撑

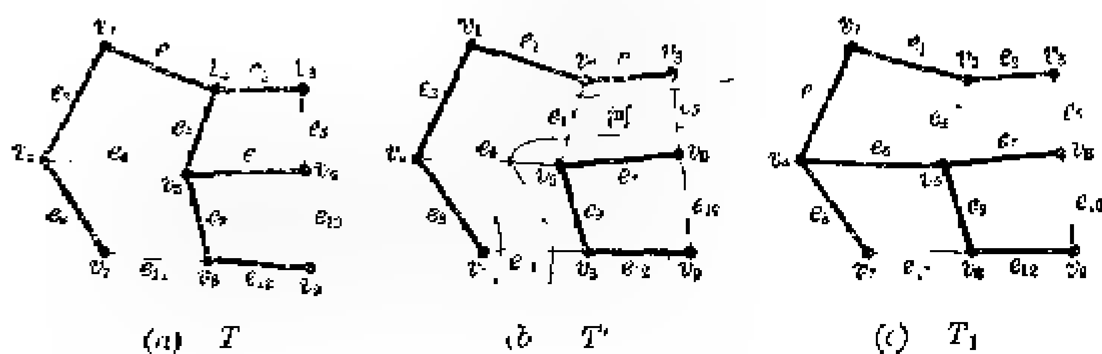
树，$e_i \in E_1$，则从 T 中去掉 e_i 所得的支撑子图 T' 不连通，并且恰好有两个连通分支.

证明 T 含有 $n-1$ 条边，所以 T' 应该含有 $n-2$ 条边. 现在来证明 T' 不连通. 用反证法. 设 T' 连通，因为 T 没有圈，T' 显然也没有圈，由定理 5.2，T' 应该有 $n-1$ 条边，但是这与刚才讲的 T' 含有 $n-2$ 条边矛盾.

由于 T' 不连通，可知 T' 至少有两个连通分支. 如果 T' 的连通分支超过了两个，那末在 T' 上增加一条边是不会一下子变成连通的(试试看，画一个有三个连通分支的图，看看加一条边能不能使它变成连通的)，但是把 e_i 加回到 T' 中去得到的 T 却是连通的. 这说明 T' 一定恰好有两个连通分支. 证完.

图 5.18(a)中的粗线表示一个支撑树 T，例如去掉 e_4 就得到(b)中的支撑子图 T'，它恰好有两个连通分支，它们的顶点集合分别是：$V_1=\{v_1, v_2, v_3, v_4, v_7\}$ 和 $V_2=\{v_5, v_6, v_8, v_9\}$.

现在就结合着图 5.18 来讲另一种从一个支撑树变换成另一个支撑树的方法. 这个方法的第一步是从支撑树 T 中去掉一条边，使 T 断裂成一个有两个连通分支的子图 T'，两个连通分支的顶点集合为 V_1 与 V_2. 然后考虑一个端点属于 V_1、另一个端点属于 V_2 的边，在图 5.18(b)中，这种边共有 4 条，就是：e_4, e_5, e_6, e_{11}. 我们就把这 4 条边组成的集合记作 $[V_1, V_2]$.（一般情况，如果 A 与 B 是顶点集合 V 的两个子集合，那末 $[A, B]$ 就表示一个端点属于 A，另一个端点属于 B 的所有边的集合）. 不难看出，把 $[V_1, V_2]$ 中的任意一条边加到 T' 中去，就可以使 T' 又变成一个连通的支撑子



(a) T　　(b) T'　　(c) T_1

图 5.18

图 T_1，而且 T_1 恰好含有 $n-1$ 条边，由树的等价定义，可以看出，T_1 是一个新的支撑树．例如，把 e_6 加到 T' 上去，得到的就是图 5.18(c) 中的支撑树 T_1．

上面讲的变换方法可以叫做"去边、架桥"变换，它的一般步骤是：

步骤 1　去边．即从支撑树 T 中任意去掉一条边，由定理 5.8，得到一个恰好有两个连通分支的支撑子图 T'．

步骤 2　架桥．设 T' 的两个连通分支的顶点集合分别为 V_1 与 V_2，从 $[V_1, V_2]$ 中任取一条边加到 T' 中去，得到一个新的支撑树 T_1．（由于增加一条边很象在河两岸陆地之间架一座桥（见图 5.18(b)）所以把这一步骤叫做架桥）．

为了证明破圈法的正确性，还要用到一个定理．

定理 5.9　设从图 G 的支撑树 T 中去掉一条边 e_i 得到 T'，T' 的两个连通分支的顶点集合为 V_1 与 V_2，又设 C 是 G 的一个圈，并且 $e_i \in C$，那末 $[V_1, V_2]$ 中一定还有一条不同于 e_i 的边也属于圈 C．

这个定理的详细证明就不讲了，留给大家作习题．不过，直观上看，这个定理是很清楚的．例如从图 5.18(b) 可以看出，如果图 G 中有一个圈 C，C 包含边 e_4［例如 $C=\{e_4, e_2,$

e_5, e_6]，想象有一个人从 v_5 开始，沿着圈 C 转一圈，最后回到 v_5．那么当他经过"桥"e_4 时，他就从河的一侧(即 V_2)走到了另一侧(即 V_1)，因此它必须从另一条"桥"走回来，即他还必须经过另一条 $[V_1, V_2]$ 的边(这里是 e_5)，才能最后又回到 v_5．

现在可以讲破圈法的证明了．和上一节一样，我们还是假定图 G 中各条边的长度都互不相等，因此，由定理 5.7，G 的最小树是唯一的，仍把它记作 T_1．

定理 5.10 用破圈法求得的一定是最小树 T_1．

证明 设 e_i 是 G 的任意一个圈 C 中的最长边，我们现在来证明：$e_i \notin T_1$．

用反证法，设 $e_i \in T_1$．按照前面讲的"去边、架桥"变换先从 T_1 中去掉边 e_i 得到 T'，T' 的两个连通分支的顶点集合仍称为 V_1 与 V_2．因为 $e_i \in C$，由定理 5.9，C 中一定还包含 $[V_1, V_2]$ 中不同于 e_i 的另一条边 e_j．现在把 e_j 加到 T' 中去，就得到一个新的支撑树 T''．从 T_1 变到 T'' 的过程中，加上的边 e_j 是 C 上的边，而去掉的边 e_i 是 C 上的最长边，所以 $l(e_i) > l(e_j)$，由此可得：$l(T'') < l(T_1)$，但这就与 T_1 是最小树矛盾了．

因为 $e_i \notin T_1$，所以把 e_i 从 G 中去掉后，得到的图 G_1 仍然包含了 T_1 的所有边，显然 T_1 也是 G_1 的最小树．如果 G_1 还有圈 C_1，那末和前面一样，C_1 上的最长边仍不属于 T_1，所以还可以去掉，也就是说，用破圈法去掉的边，都不属于 T_1，因此，在计算结束时，留下的边就恰好是 T_1 的所有边了．证完．

习　题

1．证明定理 5.9．

2．证明当 G 中有长度相等的边时，用破圈法求得的仍旧是最小树．

六、最小点基问题

1. 什么是最小点基问题

让我们先看一个例子．图6.1中画了一个有向图，它的每一个顶点 v_i 代表某篮球队的一个队员，它的弧代表的意思是：如果有一条从 v_i 出发而指向 v_j 的弧 (i, j)，就表示队员 v_i 能够通知 v_j．现在教练想要通知全体队员都来练球．请你帮教练考虑一下，他至少要通知几个队员（然后由这些队员再转告其他队员），才能使所有队员都被通知到．

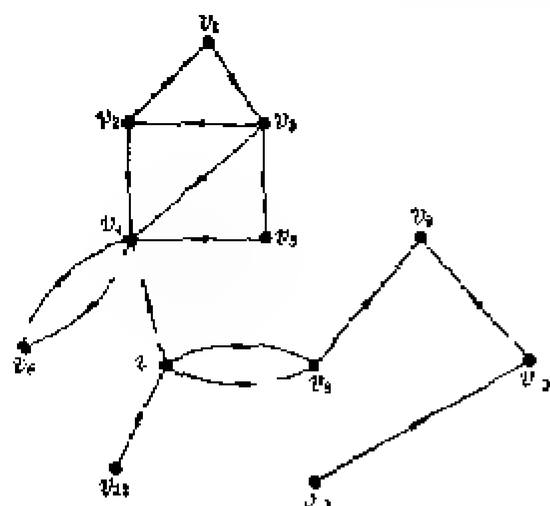


图 6.1

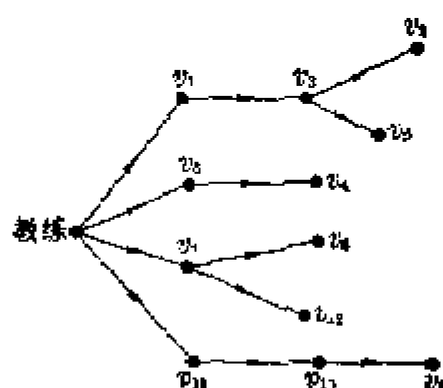


图 6.2

例如教练通知了 v_1，那末 v_1 可以再通知 v_3，v_3 又能转告 v_2、v_5、…．不难看出，如果教练通知了 v_1, v_6, v_7, v_{11}，那末，只要按图6.2画的那样做，全部队员就都能接到通知了．但

是，是不是至少通知四个队员呢？能不能再少通知几个呢？

大家试一下就会知道，通知三个人就够了，例如可以只通知 v_1，v_7 和 v_{11}，而且再少就不可能(想想看，为什么)。

上面这个例子很简单，通过简单的分析，试验就可以知道至少要通知三个人。但是如果遇到一个类似的问题，但是图中的顶点很多，那末仅仅靠试验就不行了，必须要有系统的方法。这一章的主要目的就是介绍解决这类问题的一种方法。

首先，让我们把要解决的问题明确一下。回想一下在第二章讲有向图时，我们介绍过一个概念"由 v_i 可达 v_j"(见定义 2.7)，即如果存在一条从顶点 v_i 到 v_j 的有向路，就说由 v_i 可达 v_j。为了讲起来更方便，我们再引入一个概念。

定义 6.1 设 $G=(V, A)$ 是一个有向图，v_i 与 v_j 是 G 的两个顶点，如果由 v_i 可达 v_j，就称 v_i 是 v_j 的前代，而称 v_j 是 v_i 的后代。

这个概念在我们刚才考虑的下通知问题中是很重要的，因为如果教练通知了 v_i，那末所有 v_i 的后代 v_j 就都可以间接的得到通知了，而要保证所有队员都得到通知，由教练直接通知的队员的集合 B 必须具有下面的性质："对于任意一个队员 v_j，一定存在 B 中的一个 v_i，使得 v_i 是 v_j 的前代。"或者说，B 的所有后代包括了 G 的所有顶点。按照上面的分析，可以给出下面的定义：

定义 6.2 设 $G=(V, E)$ 是一个有向图，B 是若干个顶点的集合，或者说 B 是 V 的子集。如果对于任意的 $v_j \in V$，都存在一个 $v_i \in B$，使得 v_i 是 v_j 的前代，则称 B 是一个点基。

例如按刚才讲的，在图 6.1 的图 G 中，$B_1=\{v_1, v_6, v_7, v_{11}\}$

及 $B_3=\{v_1, v_7, v_{11}\}$ 都是点基. 有了点基这个概念, 又可以提出两个图论中的极值问题了.

问题 1 求一个包含顶点最少的点基(这样的点基今后叫最小点基).

问题 2 设图 G 的每一个顶点 v_i 都对应一个非负数 a_i (a_i 叫做 v_i 的权), 现在要求一个点基, 使得它所包含的顶点对应的 a_i 之和最小(这样的点基今后叫权最小的点基).

显然, 前面讲的下通知问题可以归结为问题 1. 另外如果教练通知队员 v_i 时必须付出一定的代价 a_i(例如 a_i 可以代表教练给 v_i 打电话所需的时间), 那末, 如果教练考虑如何以最小的代价使所有队员都得到通知, 就会遇到求权最小的点基问题, 即问题 2.

注意, 如果令每一个顶点 v_i 的权 a_i 都等于 1, 那末权最小的点基就是最小点基. 因此, 问题 1 是问题 2 的一个特例.

下面就来讲问题 1 和 2 的解法. 以后将看到, 这两个问题都是较容易解决的. 不过, 为了解问题 1 与 2, 首先要学会求一个有向图的强连通分支(复习一下第二章第 3 小节)的方法, 学会了这种方法, 问题 1 与 2 就迎刃而解了.

2. 求强连通分支的方法

在第二章第 3 小节中, 我们介绍过有向图 G 的强连通分支的概念. 在那里我们也曾通过一个例子讲了求强连通分支的方法. 这种方法是: 首先任意取一个顶点 v_i, 然后把所有与 v_i 互相可达的顶点 v_j 都找出来, 这些顶点的集合 S_1 (注意 v_i

也属于S_1)组成的子图$[S_1]$(就是S_1和所有起点和终点都属于S_1的弧组成的那个子图)就是包含v_i的强连通分支．然后再取一个不属于S_1的顶点v_j，再求出与v_j互相可达的顶点集合S_2，再生成一个强连通分支$[S_2]$，…，最后就可以把所有强连通分支都求出来了．

但是，怎样从v_i求S_1呢？这个问题在第二章中没有仔细讲，现在再来讲一下．

由定义6.1，不难看出，如果顶点v_i与v_j互相可达，那末v_j既是v_i的后代(因为由v_i可达v_j)，又是v_i的前代(因为由v_j可达v_i)．因此，要求所有与v_i互相可达的顶点集合S_1，只要先求出v_i的所有后代的集合R，再求出v_i的所有前代的集合P．然后，找出所有既属于R又属于P的顶点，这些顶点就组成了集合S_1．

为了求出v_i的所有后代的集合R，也可以采用一种标号法，在这种标号法中，一旦能确定一个顶点是v_i的后代，就给它一个标号+．具体计算时，首先给v_i以标号+，这是因为v_i可达v_i自己，因此v_i本身也是v_i的后代．然后逐步扩大已标号点的范围，办法是：如果发现一个顶点v_k是已标号的，而又存在一条以v_k为起点的弧(k, j)，这时，如果v_j还没有得到标号，就可以给v_j以标号+．这样做的道理很简单，因为v_k有标号，说明它是v_i的后代，即存在一条从v_i到v_k的有向路，这条有向路接上弧(k, j)就是一条v_i到v_j的有向路(见图6.3)，因此v_j也是v_i的后代．这样不断扩大已标号顶点的范围，直到无法扩大为止，这时，所有已标号的顶点就恰好是v_i的后代的集合R了．

例如在图6.1中，要求v_2的所有后代的集合R，首先在

v_7 旁边标上 +，因为 v_7 有标号，(7, 4)是以 v_7 为起点的弧，所以 v_4 也可以得到标号，v_4 又可以使 v_6 得到标号、v_7 又可以使 v_{12} 得到标号，…，最后，在 $v_7, v_4, v_6, v_{12}, v_8, v_9$ 六个顶点得到标号后，就无法再扩大标号点的范围了，因此，R 就包含上述六个顶点.

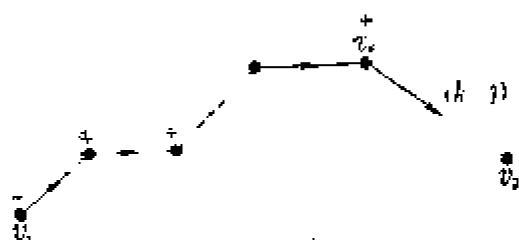


图 6.3

当一个图中顶点和弧很多时，为了使标号过程有条不紊，从而避免重复，通常在考虑一个有标号的顶点 v_k 时，应该把所有从 v_k 出发的弧(k, j)都考察一遍，如果(k, j)的终点 v_j 还没有标号，就给它标上号. 考察完以后，顶点 v_k 就可以认为已经"检查"过了，今后就不必再考虑它了(因为它再也不会使别的顶点得到标号了)，然后再取一个已标号而没有"检查"过的顶点来考虑，……，如果发现所有有标号的顶点都已经被检查过了，那末很显然，已经不可能再扩大已标号点的范围了.

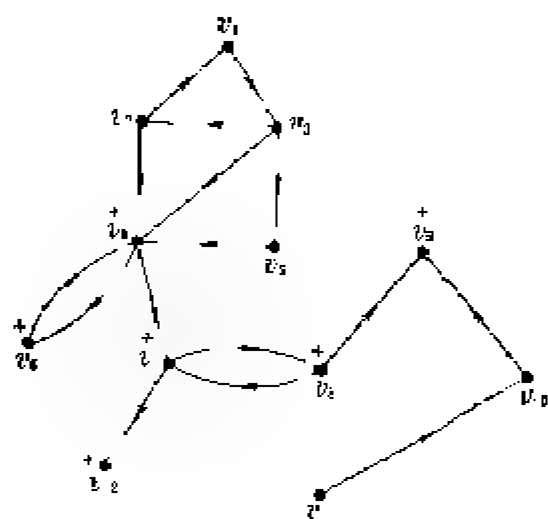


图 6.4

仍以图 6.4 为例，在给 v_7 以标号以后，只有 v_7 一个点是已标号而未检查的，考虑 v_7，这时，应该把所有从 v_7 出发的弧全考虑一遍，这里有(7, 4)、(7, 8)和(7, 12)三条，因此 v_4，v_8 和 v_{12} 都可以得到标号. 这时，v_7 已检查完了，因此

我们在 v_1 的标号＋外面画一个圈，表示它已检查过了(见图 6.5)，然后再取一个已标号而未检查过的顶点．例如取 v_{12}，没有从 v_{12} 出发的弧．因此，考虑 v_{12} 的结果没有使任何顶点得到标号．但 v_{12} 仍算已检查过了，所以在它旁边的＋外面也画一个圈，请大家继续把这个例题做下去，当然，计算的结果一定还是和图 6.4 一样的．

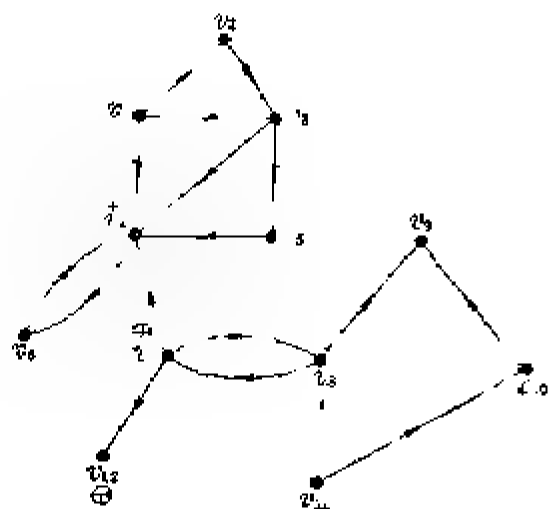


图 6.5

把上面计的总结起来，就可以得到下面的求顶点 v_i 的所有后代的集合 R 的计算方法了：

步骤 1 给 v_i 以标号＋，v_i 是已标号而未检查的顶点．

步骤 2 看看是否有已标号而未检查的顶点，如果没有，计算结束，所有已标号的顶点就组成集合 R．如果有，任取一个这样的顶点 v_k，转入步骤 3．

步骤 3 找出以 v_k 为起点的所有弧 (k,j)，一条一条地考虑这些弧，如果 (k,j) 的终点 v_j 没有标号，就给 v_j 以标号＋，v_j 成为已标号而未检查的顶点．这些弧都考虑过以后，v_k 成为已检查过的，在它旁边的＋外面画一个圈，转回步骤 2．

如果想求顶点 v_i 的所有前代的集合 P，可以采取与上面完全相似的方法．步骤 1 和 2 是一样的，唯一的差别是步骤

3. 本来，在取定了 v_k 以后，考虑的是以 v_k 为起点的弧 (k, j)，现在则应该改为考虑以 v_k 为终点的弧 (j, k)，如果这条弧的起点 v_j 还没有标号，就应该给它以标号。另外，为了在一张图上同时求 v_i 的所有前代与后代，那末，在求前代时，可以用"－"作为标号。

仍以图 6.1 中的图为例，现在来求 v_7 的前代的集合 P，在给 v_7 以标号"－"以后，考虑以 v_7 为终点的弧，这里只有(8,7)一条，因此 v_8 也可以得到标号，但是，别的顶点再也得不到标号了，因此 $P=\{v_7, v_8\}$。

在图 6.6 中，标有＋的顶点是 v_7 的后代，标有－的顶点是 v_7 的前代，因此，很显然，既标有＋又标有－的顶点就是与 v_7 互相可达的顶点了。从图6.6可以看出，这种顶点只有 v_7 与 v_8 两个，因此，与 v_7 对应的 $S_1=\{v_7, v_8\}$，而图 6.7 中所画的就是 S_1 生成的子图 $[S_1]$ 了。

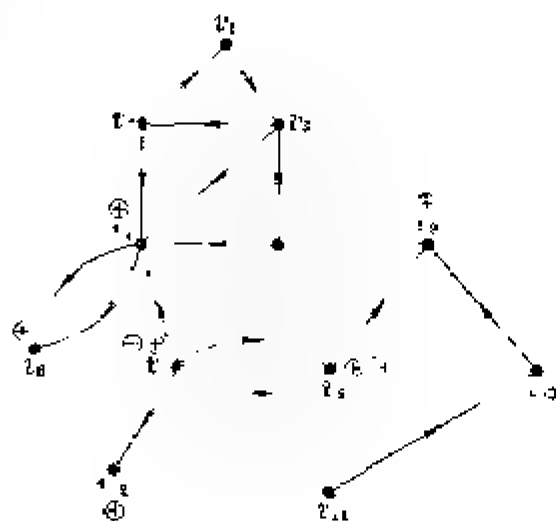


图 6.6

在求出了一个强连通分支 $[S_1]$ 以后，应该看看，是不是有不属于它的顶点。有的话，就再任意取一个不属于 S_1 的顶点 v_j，再求出包含 v_j 的强连通分支 $[S_2]$；然后再取一个既不属于 S_1 又不属于 S_2 的顶点 v_k，…，直到每一个顶点都属于一个强连通分支为止。



图 6.7

例如在图 6.6 中，取不属于 S_1 的顶点 v_9，不难算出，v_9

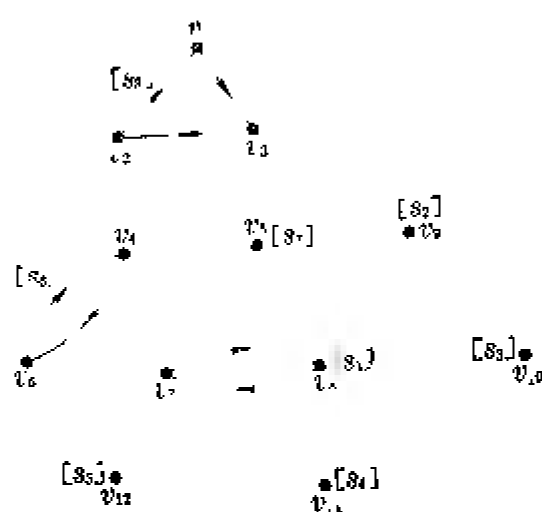


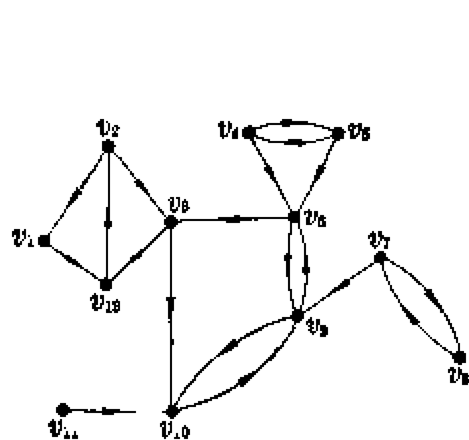
图 6.8

的后代集合 $R=\{v_9\}$，前代的集合 $P=\{v_7, v_8, v_9, v_{10}, v_{11}\}$，所以，$S_2=\{v_9\}$，当然它生成的子图也只包含 v_9 一个顶点，而不包含任何弧. 继续做下去，最后可看出，图6.6中的有向图共有八个强连通分支，它们分别是由：$S_1=\{v_7, v_8\}$，$S_2=\{v_9\}$，$S_3=\{v_{10}\}$，$S_4=\{v_{11}\}$，$S_5=\{v_{12}\}$，$S_6=\{v_4, v_6\}$，$S_7=\{v_5\}$，$S_8=\{v_1, v_2, v_3\}$生成的(见图6.8).

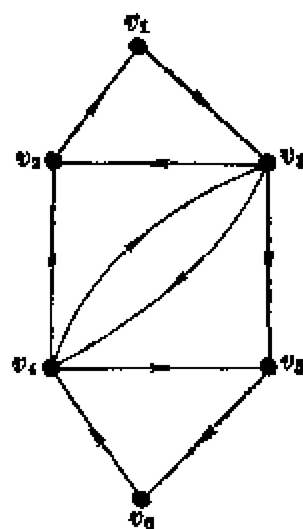
习 题

1. 写出求一个顶点 v_i 的所有前代的集合 P 的计算步骤，并画出框图.

2. 求出下面两个有向图的所有强连通分支：



(a)



(b)

3. 证明本节所讲的求强连通分支的计算方法是有效的.

3. 问题1与2的解法

现在可以回过头来讲第1小节中提出的两个问题的解法了. 不过在讲以前,我们还要介绍一个概念.

定义6.3 设 $[S_i]$ 是有向图 G 的一个强连通分支, 如果在 G 中,不存在终点属于 $[S_i]$ 而起点不属于 $[S_i]$ 的弧, 就称 $[S_i]$ 为最高的强连通分支.

现在就用这个定义来判断一下. 图6.8的八个强连通分支中,哪几个是最高的. 先看 $[S_1]$, 从图6.9中不难看出, 终点属于 $[S_1]$ 的弧只有两条, 即(7, 8)与(8, 7),它们的起点也都属于 $[S_1]$. 也就是说,找不到终点属于 $[S_1]$ 而起点不属于 $[S_1]$ 的弧,因此 $[S_1]$ 是最高的. 再看 $[S_2]$, $[S_2]$ 只包含一个顶点 v_9, 终点属于 $[S_2]$ 的弧有(8, 9)与(10, 9)两条. 但它们的起点不属于 $[S_2]$, 因此 $[S_2]$ 就不是最高的. 用同样的办法,可以看出,在八个强连通分支中, $[S_1]$, $[S_4]$ 和 $[S_8]$ 是最高的,而其他五个都不是最高的.

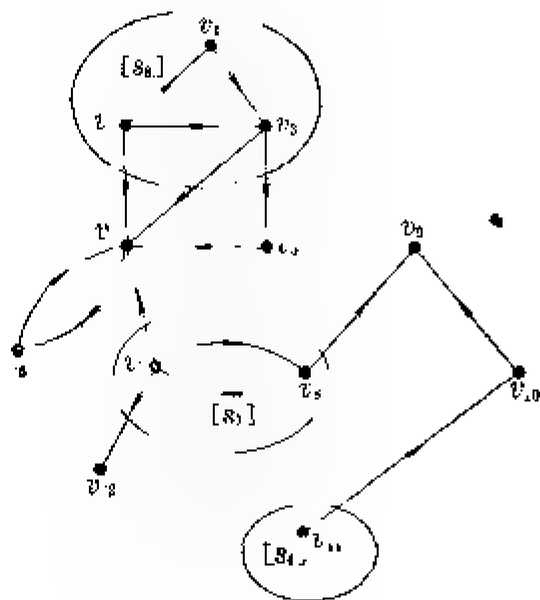


图 6.9

讲到这里,需要的准备工作都已做好, 可以讲问题1与2的解法了.

问题1的解法如下.

步骤1 找出图 $G=(V, A)$ 的所有强连通分支.

步骤2 从强连通分支中找出所有最高的强连通分支.

步骤3 从每个最高的强连通分支中任意取一个顶点，组成的顶点集合 B 就是 G 的一个最小点基.

按照上面讲的三个步骤来解决前面讲的教练通知问题就很容易了. 上一节已经求出了图6.1中的图的所有强连通分支(见图6.8). 刚才又已经决定了有三个强连通分支是最高的，就是$[S_1]$、$[S_4]$和$[S_8]$. 因此，只要再按步骤3，从$[S_1]$、$[S_4]$和$[S_8]$中各任意取一个顶点，例如从$[S_1]$中取 v_8, $[S_4]$中取 v_{11}, $[S_8]$ 中取 v_3. 得到的 $B_1=\{v_3, v_8, v_{11}\}$ 就是一个最小点基了. 另外, $B_2=\{v_2, v_8, v_{11}\}$, $B_3=\{v_1, v_7, v_{11}\}$, ……也都是最小点基，因此教练可以只通知 v_3, v_8 和 v_{11}; 也可以象第1小节中讲的那样，通知 v_1, v_7 和 v_{11}. 不难看出，这个问题一共有六个不同的最小点基.

分析一下上面的三个计算步骤，就会看出这些步骤都是比较容易做的，稍为麻烦些的恐怕还是步骤1. 但是，要证明用这三个步骤求出来的一定是最小点基，也就是说，证明上面讲的方法是正确的，还要化一些功夫，首先让我们证明两个引理.

引理6.1 设 $[S_i]$ 是一个最高的强连通分支，又 v_j 属于 $[S_i]$ 而 v_k 不属于 $[S_i]$，那末 v_j 不会是 v_k 的后代(顺便说一下，这个引理也说明了 $[S_i]$ 确实在某种意义下是最高的).

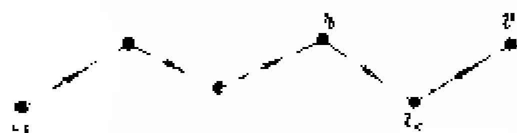


图6.10

证明 用反证法. 如果 v_j 是 v_k 的后代，即存在一条从 v_k 到 v_j 的有向路 p (见图6.10中的示意图)，现在假

设从 v_k 出发，沿着 p 向前走时，第一个遇到的属于 $[S_i]$ 的顶点是 v_c，也就是说 v_c 前面的一个顶点 v_b 不属于 $[S_i]$，但这时弧 (v_b, v_c) 将是一条终点属于 $[S_i]$ 而起点不属于 $[S_i]$ 的弧，这就与 $[S_i]$ 是最高的强连通分支矛盾了．证完．

现在，让我们结合着教练下通知问题来想想看，引理6.1说明了什么问题？不难看出，它说明了，如果 v_j 是一个属于最高强连通分支 $[S_i]$ 的队员，那末任何不属于 $[S_i]$ 的队员 v_k 是无法通知到他的，因此，教练至少要直接通知 $[S_i]$ 中的一个队员．用图论中的话来说，应该把上面的结论说成 如果顶点集合 B 是一个点基，那末每个最高连通分支 $[S_i]$ 至少有一个顶点要属于 B．而由此显然又可以得到一个结论．就是：

引理 6.2 设图 G 有 t 个最高强连通分支，那末 G 的每一个点基 B 至少包含 t 个顶点．

由引理6.2可以看出，如果能找到一个点基 B 恰好包含 t 个顶点，B 就一定是最小点基了．因此，要证明上面的计算方法是正确的，只要能证明："从 t 个最高的强连通分支中任意各取一个顶点组成的集合 B 一定是一个点基"就可以了．下面就来证明这一结论．

定理 6.1 从 G 的每一个强连通分支中任意各取一个顶点组成的集合 B 是 G 的一个点基．

证明 要证明的是，对于 G 的任一顶点 v_j，一定可以从 B 中找出一个 v_i，使得 v_i 是 v_j 的前代．

首先设 v_j 属于某一个最高的强连通分支 $[S_k]$，那末按照 B 的造法，$[S_k]$ 中一定有一个顶点属于 B．设它为 v_i，v_i 与 v_j 都属于 $[S_k]$，因此是互相可达的，当然由 v_i 可达 v_j，即在 B 中，已经找到了一个 v_i 是 v_j 的前代了．

再设 v_j 属于强连通分支$[S_k]$，而$[S_k]$不是最高的．那末一定存在一条终点属于$[S_k]$而起点不属于$[S_k]$的弧(v_y, v_x)，设 v_y 属于另一个强连通分支$[S_l]$．如果$[S_l]$是最高的，就好了，因为这时$[S_l]$中一定有一个顶点 v_i 属于 B（见图 6.11），因为由 v_i 可达 v_y（它们都属于$[S_l]$），由 v_y 可达 v_x，而由 v_x 又可达 v_j（都属于$[S_k]$），因此由 v_i 可达 v_j，即又已证明了在 B 中找到了一个 v_i，它是 v_j 的前代．但是如果$[S_l]$仍不是最高的，怎么办呢？这也不要紧，因为这时又可以找到一条终点属于$[S_l]$而起点不属于$[S_l]$的弧(v_a, v_b)，设 v_a 属于强连通分支$[S_p]$，如果$[S_p]$是最高的，和前面相似可证，$[S_p]$中属于 B 的 v_i 就是 v_j 的前代；如果$[S_p]$还不是最高的，就再找一个$[S_q]$，…，这样找下去，一直到得到一个最高的强连通分支$[S_r]$为止，和前面相似可以证明，$[S_r]$中属于 B 的顶点 v_i 是 v_j 的前代（见图 6.12，图中的粗弧表示从 v_i 到 v_j 的有向路）．

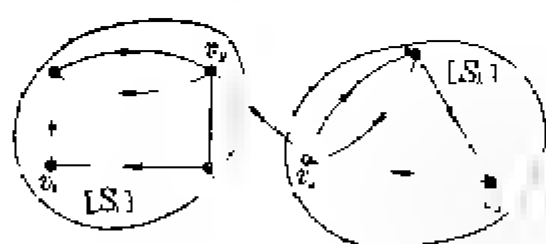


图 6.11

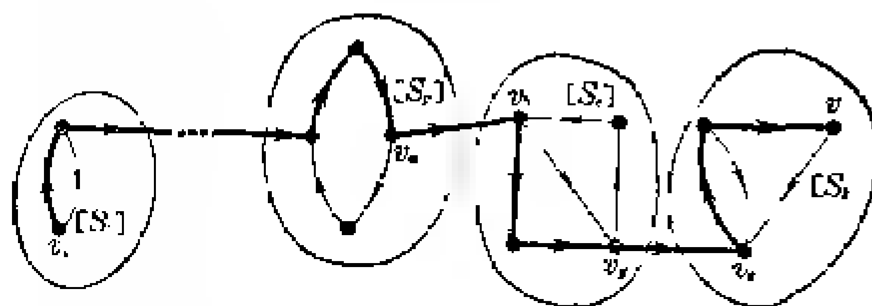


图 6.12

这里还有一个问题，就是象上面讲的那样从$[S_k]$往前找到$[S_l]$，再往前找到$[S_p]$，…，是不是一定会找到一个最高的强连通分支呢？答案是肯定的，因为在图 G 中，只有有限个强

连通分支，而且可以证明，从$[S_k]$到$[S_l]$，再到$[S_t]$……，在这个过程中出现的所有强连通分支都互相不同．为什么呢？因为如果有一个强连通分支出现了两次，例如象图6.13中画的那样，从$[S_k]$开始找，先得到$[S_l]$，然后又得到$[S_p]$，$[S_q]$，从$[S_q]$又得到了$[S_l]$，也就是说，$[S_l]$出现了两次，这时就有矛盾了，因为不难看出，$[S_l]$的任一顶点v_i与$[S_p]$（或$[S_q]$）的任一顶点v_j将是互相可达的，这就与它们属于不同的强连通分支矛盾了．

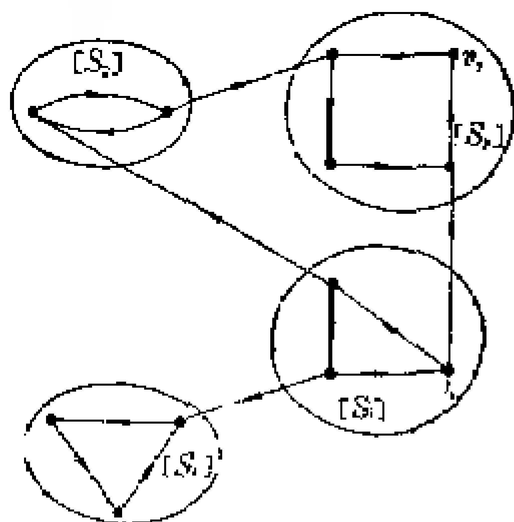


图 6.13

从上面讲的可以看出，不管那一种情况，对于G中的顶点v_j，总可以在B中找到一个v_i，使v_i是v_j的前代，这就证明了B是G的点基．定理6.1证完．

到此，求最小点基的方法的正确性就证完了．

最后，再讲一下问题2的解法．表面上看，似乎问题2要比问题1复杂，其实不然，这两个问题的复杂程度是差不多的．要求一个权最小的点基，也需要进行三个步骤，其中步骤1和2跟求最小点基完全一样，而步骤3则要改为：

步骤3′ 从每一个最高的强连通分支中取一个权最小的顶点，组成的顶点集合B就是G的一个权最小的点基．

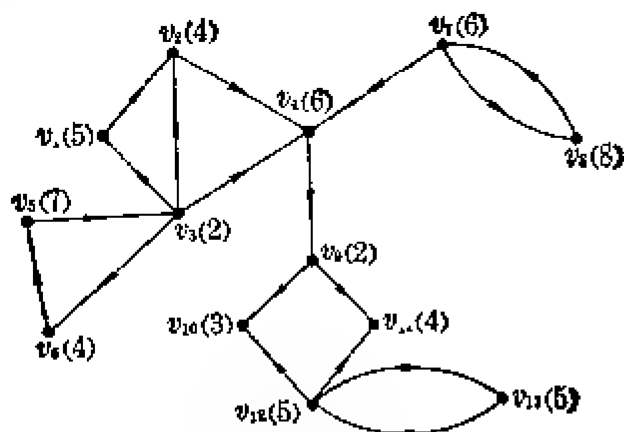
至于这个方法的正确性，这里就不写了，留给大家作为习题吧．

习　　题

1、为第2小节问题2中的两个有向图各找一个最小点基。

2、证明解问题2的方法的正确性。

3、为下面的有向图 G 找一个权最小的点基（顶点旁边括号中的数字代表权）。



4、证明解问题1与2的方法都是有效的。

七、最小树形图问题

1．从渠道设计问题谈起

第五章讲最小树问题时，曾说过，最小树问题也可以用来解决渠道设计问题．但是这种说法仅仅适用于水源以及需要灌溉的各块地的高低都一样的情况，如果水源和各块地的地势有高有低，问题就比较复杂了．

例如在图 7.1(a) 的有向图 D 中，v_1 表示水源，其他顶点表示需要灌溉的地，各条弧代表可以选来修建渠道的线路，从弧上的箭头可以看出地势的高低，例如有一条从 v_1 到 v_2 的弧，这就表示 v_1 比 v_2 高．水可以从 v_1 流到 v_2 去．又如既有从 v_1 到 v_3 的弧又有 v_3 到 v_1 的弧，这说明 v_1 的地势和 v_3 一样高，水既可以从 v_1 流到 v_3，也可以从 v_3 流到 v_1．现在需要

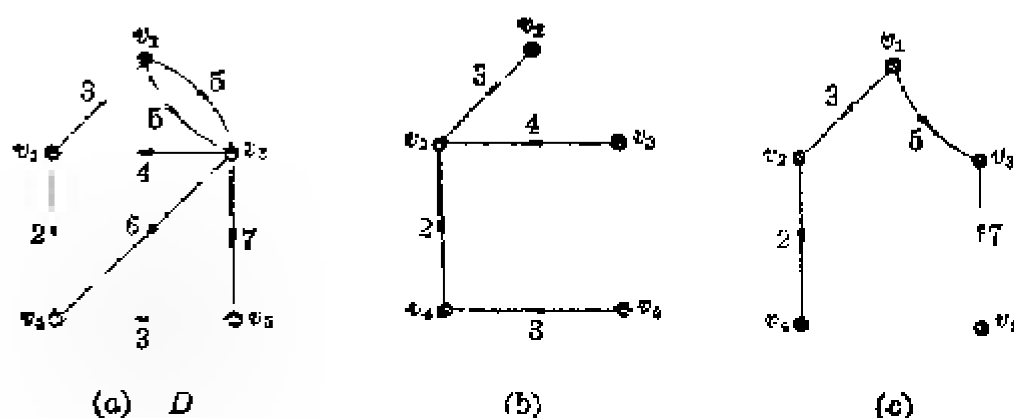


图 7.1

解决的问题是 应该选择哪几条路线修建渠道才最节省?

如果不考虑弧的方向，用第五章的办法(例如用破圈法)求最小树，那末求得的将是(b)，它的各条弧的总长是12，但是按这个图来修渠道是不行的，因为v_1的水流不到v_3与v_5去！要使各块地都能得到灌溉而且总长度又最小的设计方案应该是图7.1(c)中画的那样。虽然它的各条弧的长度的和是17，比(b)的大，但是这样设计v_1的水能够流到各块地上去的。

那末图7.1(c)是怎样求出来的? 怎么知道它是最节约的呢? 这些问题就是本章要解决的主题。不过，为了弄清楚这些问题，必须先弄清楚什么是树形图。

分析一下图7.1(b)、(c)有什么特点。显然，它们的公共特点是：如果不考虑弧的方向，它们都是树。用第二章中的名词说，则应该说成：它们的相伴无向图是树。但是，作为一个渠道设计的方案来说，单单具有这个性质还不够，它至少还需要一个性质，就是：水源的水能流到各块地上去。下面定义7.1中的性质(2)就是从这种分析中抽象出来的。

定义7.1 设$H=(V, A)$是一个有向图，如果它具有下述性质：

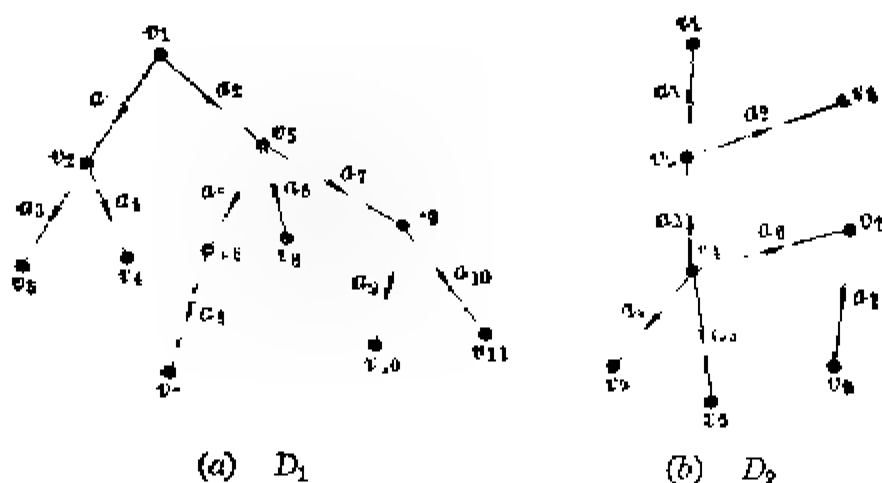
(1) 它的相伴无向图是一个树；

(2) 存在一个顶点v_i，它具有下述性质：对于V中任意其他的顶点v_j，存在从v_i到v_j的有向路。则称H是一个树形图，而顶点v_i称为这个树形图的根。

与支撑树相似，我们也可以定义支撑树形图。

定义7.2 设$D=(V, A)$是一个有向图，$H=(V, A_1)$是D的支撑子图，并且H是一个树形图，则称H是D的支撑树形图。

例如图 7.2(a)、(b) 中画的 D_1 与 D_2 都是树形图的例子，D_1 的根是 v_1，D_2 的根是 v_4（请大家检查一下，定义 7.1 中的 (2) 是否成立）。另外，不难看出，图 7.1(c) 中的图是一个树形图（从而它是 (a) 中的有向图 D 的支撑树形图），而 (b) 不是树形图。仔细考虑一下，就可以看出，(c) 中的图代表的渠道设计方案所以是可行的，原因就在于它是一个树形图，而 (b) 所以不可行也就是因为它不是树形图。由此可见，选择最节约的渠道设计方案问题，可以归结为下述极值问题：设给定了有向图 $D=(V, A)$，它的每条弧都有一个非负的长度，现在要从 D 的所有以顶点 v_1（相当于水源）为根的支撑树形图中，找出弧的总长度最小的支撑树形图来。



(a) D_1　　(b) D_2

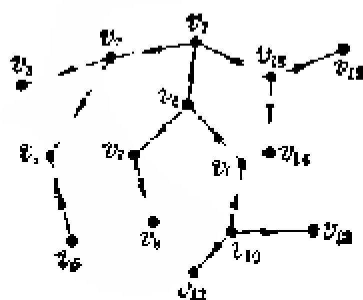
图 7.2

和第五章中讲最小树一样，如果 H 是 D 的支撑树形图，我们就把 H 中各条弧的长度相加的和记作 $l(H)$，并且把它叫做 H 的长度。而把长度最小的支撑树形图叫做最小树形图。因此，上面讲的那个极值问题就叫做最小树形图问题。当然，最小树形图问题不仅可以用来解决渠道设计问题，还可以用来解决其他实际问题，例如通讯网或交通网的设计等等。

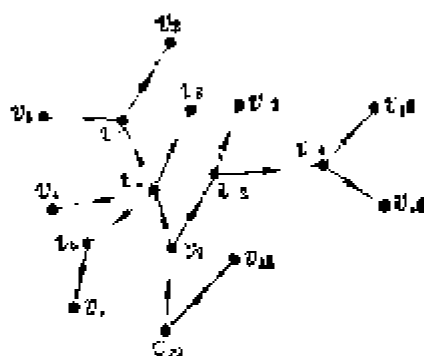
本章的目的就是介绍求有向图的最小树形图的方法. 总起来说, 求最小树形图要比求最小树麻烦一些, 不过只要稍为耐心些学习, 还是能学懂的. 有些比较难懂的证明本书就不仔细写了。

习　题

判别一下, 下面的两个有向图是不是树形图? 如果是的, 指出它的根是哪一个顶点.



(a)



(b)

2. 树形图的等价定义

你做过上一节的习题 1 了吗? 做过的话, 就会发现, 要判断一个有向图是不是树形图还是比较麻烦的. 特别是判断定义 7.1 中的性质(2)是不是成立比较麻烦. 当然, 如果应用上一章的概念仔细分析一下, 就可以看出, 性质(2)相当于: '存在一个顶点 v_i, 使得图中所有顶点都是 v_i 的后代.' 因此, 要判断性质(2)是不是成立, 可以用上一章讲的求顶点 v_i 的后代集合的办法, 一个一个地求顶点 $v_1, v_2, \cdots, v_n$ 的后代集合; 如果发现有一个顶点 v_i, 它的后代集合包含了 D 的所有顶

点，就可以肯定性质(2)成立了，不过这样做仍是相当麻烦的。

有没有更简单的办法来判断一个有向图是不是树形图呢？有的，用下面讲的树形图的等价定义来判断就要简单得多。

定义 7.2 设 $D=(V, A)$ 是一个有向图，如果它具有下面的性质：

(1) 它的相伴无向图是一个树.

(2') 存在一个顶点 v_i，它不是任何弧的终点，而 V 的其他顶点都恰好是唯一的一条弧的终点. 就称 D 是以 v_i 为根的树形图.

我们给树形图下了两个不同的定义了. 一般说来，在数学中，同一个概念不能下两个不同的定义. 不过，如果能接着就证明这两个定义是等价的，就不要紧了. 下面，我们通过两个引理来证明定义 7.1 与 7.2 是等价的.

引理 7.1 设有向图 $D=(V, A)$ 具有定义 7.1 中的性质(1)与(2)，则 D 也具有定义 7.2 中的性质(2').

证明 设 v_i 是性质(2)中所说的那个顶点. 那末，首先，不难看出，对于 D 中不同于 v_i 的顶点 v_j 来说，在 D 中至少存在一条弧以 v_j 为终点，这是因为由于定义 7.1 的性质(2)成立，D 中存在从 v_i 到 v_j 的有向路 p，p 的最后一条弧显然是以 v_j 为终点的.

其次，因为 D 的相伴无向图是树，因此 D 共有 $n-1$(n 是 D 的顶点的个数)条弧。

有了上面两点，要证明性质 (2') 成立就很容易了. 大家可以这样想，让 D 的每一个顶点代表一个小孩，每一条弧代表

一个苹果，如果弧 a_k 以 v_i 为终点，我们就认为 a_k 这个苹果属于小孩 v_i 所有。那末前面已经证明了：首先，除了 v_i 以外，其他的小孩（注意，一共是 n 个顶点，即 n 个小孩，除了 v_i 以外，还有 $n-1$ 个）每人都至少有一个苹果，其次，总共只有 $n-1$ 个苹果。想想看，可以得出什么结论？当然是除了 v_i 以外，每个小孩都恰好有一个苹果，而 v_i 呢，他一定没有苹果。而这就正好说明了：没有弧以 v_i 为终点，面对于不同于 v_i 的顶点 v_j，恰好有一条弧以 v_j 为终点，而这就是性质 (2′)。引理 7.1 证完。

引理 7.2 设有向图 $D=(V, A)$ 具有定义 7.2 中的性质 (1) 与 (2′)，则 D 也具有定义 7.1 中的性质 (2)。

证明 设 v_i 是性质 (2′) 中所说的那个不是任何弧的终点的那个顶点。现在来证明对于任何不同于 v_i 的顶点 v_j，在 D 中存在从 v_i 到 v_j 的有向路。

对于顶点 v_j，由于性质 (2′) 成立，一定有一条以 v_j 为终点的弧 a_k，设 a_k 的起点是 v_l，如果 v_l 就是 v_i，那末从 v_i 到 v_j 的有向路已经找到了，即 $\{v_i, a_k, v_j\}$。如果 $v_l \neq v_i$，那末又存在以 v_l 为终点的弧，…，这样往上找，最后一定会得到一条弧，它的起点是 v_i，而这时，从 v_i 到 v_j 的有向路就找到了。

你是否感到好象在什么地方见过类似的证明方法？是的，在上一章第 3 小节定理 6.1 的证明中用过类似的方法。和那里相似，应该证明在从 v_j 开始，往前找的过程中，所有出现的顶点都互相不同，而且只要 v_i 不出现，就一定可以不断的找下去，而这就可以证明 v_i 最终一定要出现的了。不过，关于这些，这里就不仔细写了，请大家自己写一下整个这个引理的严格证明吧。引理 7.2 证完。

定义7.1与定义7.2的等价性已经证明完了，试一下就会知道，用定义7.2来判断上一节习题1中的图是不是树形图比较简单(例如对于(a)中的图，一发现v_4是两条弧的终点就马上可以肯定它不是树形图了)。当然，从理论上来分析，也可以看出性质(2')比性质(2)易于检查的原因，因为性质(2')是一个"局部"性质，要检查它是不是成立，只要在每一个顶点的局部来检查就可以，即看一下有几条弧以这个顶点为终点，而要检查性质(2)是不是成立，就要从一个图的整体上来看，因此就比较麻烦了。

定义7.2中的性质(1)还可以修改，即树形图还可以有下面的等价定义：

定义7.3 设$D=(V, A)$是一个有向图，如果它具有下面的性质：

(1') D不包含有向圈。

(2') 存在一个顶点v_1，它不是任何弧的终点，而V的其他顶点都恰好是唯一的一条弧的终点。

则称D是以v_1为根的树形图。

不难看出，用定义7.3来检查一个图是不是树形图比定义7.2更简单。至于定义7.3与定义7.2的等价性，也留给大家做习题。

习　题

1 详细写出引理7.2的证明。

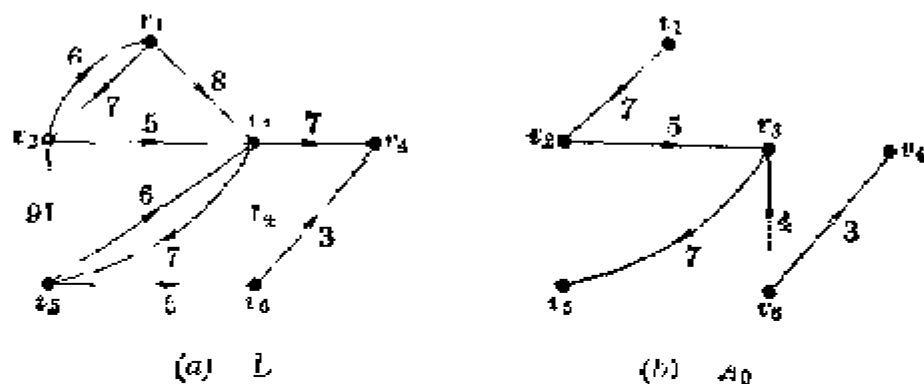
2 利用上一章的结果给出引理7.2的另一种证明方法（先证明在定义7.2中的(1)(2)成立时，Γ的每一个强连通分支都只包含一个顶点，然后证明只有由v_1组成的强连通分支是最高的。

3. 证明定义7.2与定义7.3是等价的。

3. 关于最小树形图求法的初步讨论

这一节开始讲怎样求一个有向图 $D=(V, A)$ 的以顶点 v_i 为根的最小树形图。为确定起见，下面都假设指定为根的顶点是 v_1。

要找 D 的一个支撑树形图，主要就是确定它包含哪 $n-1$ 条弧。由上一节最后讲的树形图的等价定义 7.3 可知，一个支撑树形图的 $n-1$ 条弧必须分别以 $v_2, \cdots, v_n$ 为终点。这就使人很自然的想到一种求最小树形图的方法了，那就是：从所有以 v_2 为终点的弧中取一条最短的，再从以 v_3 为终点的弧中取一条最短的，…，最后，从以 v_n 为终点的弧中取一条最短的。这样就得到了 $n-1$ 条弧，我们称这 $n-1$ 条弧组成的集合为 A_0，A_0 与 D 的顶点集合 V 合在一起就形成 D 的一个支撑子图 (V, A_0)（今后，为简单起见，有时就把弧集 A_0 叫做支撑子图 A_0，遇到这种说法时，我们应理解成是指由 A_0 与 V 组成的支撑子图 (V, A_0)），它显然具有定义 7.3 中的性质 (2')，因为 v_1 不是 A_0 中任何弧的终点，而其他顶点都恰好是 A_0 的唯一的一条弧的终点。而且不难看出，A_0 还是所有具



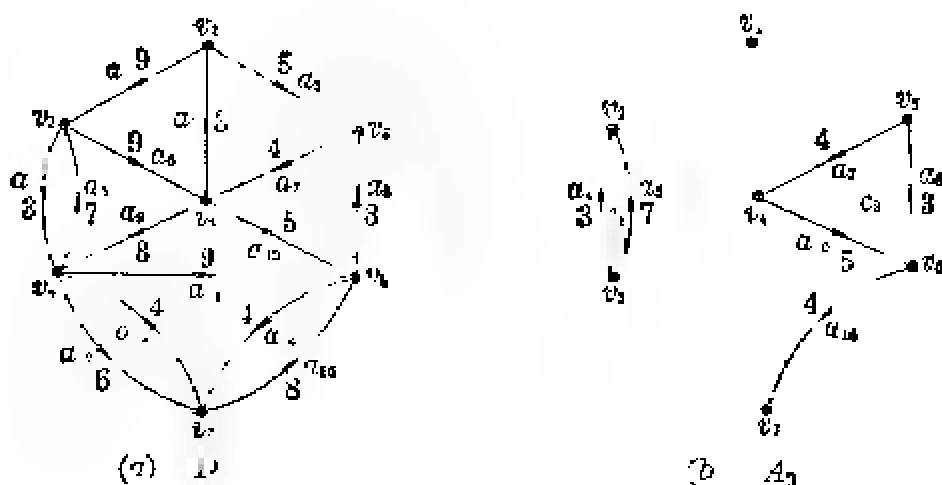
(a) D　　(b) A_0

图 7.3

有性质 (2′) 的支撑子图中，弧的总长度最短的一个. 因此如果碰巧 A_0 还具有定义 7.3 中的性质 (1′)，即它不包含有向圈. 那末 A_0 就是 D 的一个支撑树形图，而且很显然，它一定也就是 D 的以 v_1 为根的最小树形图了.

举个例说，对于图 7.3(a) 中的有向图 D，分别取以 v_2, v_3, $\cdots$, v_6 为终点的五条最短的弧，就得到图 7.3(b) 中的支撑子图，它不包含有向圈，因此它就是 D 的最小树形图了. 另外，这一章第 1 小节中的图 7.1(c) 其实也是用这种方法求出来的.

当然，用上面的办法只是"碰巧"才能得到最小树形图. 因为在大多数情况下，用上述办法得到的 A_0 不一定满足定义 7.2 中的性质(1′)，即它不一定是树形图. 例如对于图 7.4(a) 中的有向图 D 来说，用上面办法得到的支撑子图 A_0 就是插图 7.4(b) 中的图，它有两个有向圈，因此，A_0 不是 D 的支撑树形图.



(a) D　　(b) A_0

图 7.4

在 A_0 不具有定义 7.3 中的性质 (1′) 时怎么办呢? 是不是把 A_0 全部不要了再另外设法重新找最小树形图呢? 当然

不必要．我们可以这样想，例如某学校要从10个区招10个学生，规定各个区招一个，并且要求男、女学生各占一半，于是我们采取了下述办法，首先，从每个区中选一个最优秀的学生，如果这10个学生正好5个男生5个女生，那就好了，就招这10个．如果是6个男生，4个女生呢？当然没有必要把这10个学生全部退回去，再重新考虑，而应该以这10个学生为基础，稍加调整而得出最后的录取名单．对于A_0来说也一样，当它不具有性质(1′)时，我们还是希望能以它为基础，稍加修改而得到要求的最小树形图．

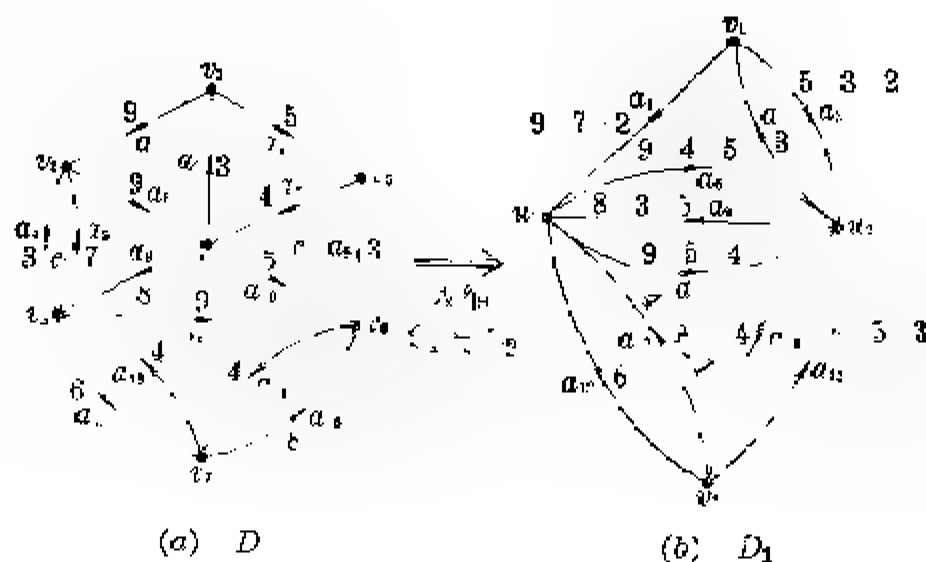
怎么修改呢？我们就结合着图7.4(b)来讲，A_0有两个有向圈C_1与C_2．对修改的目的就是使这些有向圈"破"掉，当然在破的同时，还应注意保持住定义7.3中的性质(2′)成立，并且要使总长的扩大尽量地小．那末怎样破才能达到这些目的呢？拿C_1来讲，C_1包含两条弧a_4与a_5，为了把C_1破掉，这两条弧至少有一条要从A_0中去掉，那末去谁呢？有的同志也许会脱口而出："当然去a_5，因为它的长度是7，而a_4的长度是3"．其实这个想法是不对的，因为为了保持定义7.3中的性质(2′)成立，在去掉弧a_5后，必须再取一条以v_2为终点的弧补进来，从图7.4(a)可以看出，除了弧a_5以外，以v_2为终点的弧还有a_1，它的长度是9．因此去掉a_5，补上a_1，弧的总长度将增加9−7=2．而如果去掉弧a_4呢，应该再补一条以v_3为终点的弧，这里可以考虑的有a_9与a_{13}两条，当然取较短的a_{13}，它的长度是4，因此用弧a_{13}换a_4，弧的总长度只增加4−3=1．经过这样一比较，就可以看出，实际上去掉弧a_4要比去掉弧a_5划算．

用相似的办法，可以算出对于圈C_2来说，最划算的是去

掉弧 a_8，换进弧 a_3，弧的总长度将增加 5－3＝2.

我们还可以采取另外一种办法来决定每个有向圈中哪一条弧应该去掉．这种新办法的背景其实就是前面两段讲的那些内容，但是它包含了一种很好的新想法——收缩，并且计算起来也更加有效．

我们要介绍的新办法是：在发现 A_0 中包含有向圈时，我们就把原来的有向图 D 变一下．变的办法是把 A_0 中的每一个有向圈都"收缩"成一个顶点．拿图 7.4(a) 中的 D 来说，应该变成图 7.5(b) 中的 D_1．即把有向圈 C_1 和 C_2 分别收缩成 D_1 中的顶点 u_1 和 u_2，u_1 和 u_2 今后称为收缩点．D 中的弧，如果两个端点都属于 C_1 或都属于 C_2，就被收缩掉了，其他的弧都保留着．



(a) D　　(b) D_1

图 7.5

有意思的是，在从 D 变为 D_1 的过程中，弧的长度要变化．变化的规律是：凡是终点不是收缩点的弧，它的长度不变，而所有以收缩点为终点的弧，长度就要变化．具体到图 7.5(b)，就是所有以 u_1 和 u_2 为终点的弧的长度要变化．在

图7.5(b)上，我们已经标出了变化后的弧长。例如弧a_1，我们标出了它的长度应该是9－7，即2。这里9是a_1原来在D中的长度，而7是有向圈C_1中与a_1有相同终点v_2的弧a_5的长度（见图7.6）。为什么这样改呢？前面讲过，如果把C_1中的弧a_5去掉，而补上弧a_1，弧的总长要将增加9－7＝2，这样一想就清楚了。原来弧a_1在D_1中的长度就是考虑用弧a_1去代替C_1中的a_5（要加进去的是a_1，那么为了保持定义7.3中的性质(2')成立，去掉的必须是C_1中与a_1有相同终点的弧，即a_5）时弧长的增加量。其他以v_1或v_2为终点的弧的长度也都是按照这个规律变化的。

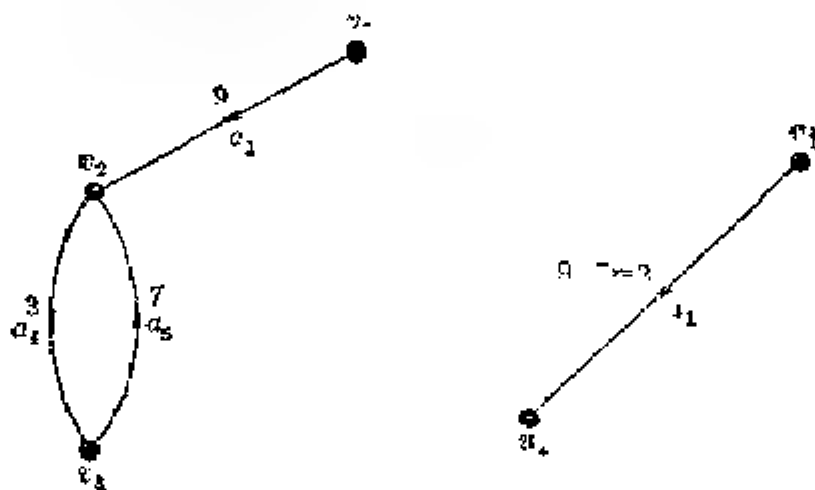


图 7.6

有了D_1，就可以用另一种办法很快地决定怎样对A_0进行修改最划算了。那就是：只要先看一下，D_1中以v_1为终点的弧中，哪一条最短。例如这里是a_{13}最短，它的长度是1，这就说明用a_{13}去代替C_1中的一条弧（被替换的一定是C_1中与a_{13}有相同终点的弧，即a_4）最划算。由同样的道理，D_1中以v_2为终点的弧中，最短的是a_3，它的长度是2，因此，要破掉圈C_2，最划算的办法是用弧a_3替换a_8，这也和前面计算的结

果 效.

有了上面这些准备，再来学习下一节要讲的求最小树形图的计算方法，就不会感到困难了.

4. 求最小树形图的计算方法

现在就来系统地介绍一种求有向图 $D=(V, A)$ 以 v_1 为根的最小树形图的方法，这种方法就是以上一节讲的"取指向每一个顶点的最短弧"和"收缩有向圈"为基础的，它的具体计算步骤如下.

步骤 1 令 $D_0=D$，分别选取以 $v_2, v_3, \cdots, v_n$ 为终点的最短弧，可能出现两种情况：

情况 1 发现一个顶点 $v_j(v_j\neq v_1)$ 不是 D_0 中任何弧的终点，这时计算结束，因为显然 D_0 中不存在以 v_1 为根的支撑树形图；

情况 2 情况 1 不出现，这时，可以得到 $n-1$ 条弧，分别以 $v_2, v_3, \cdots, v_n$ 为终点，把这些弧组成的集合叫做 A_0，转入步骤 2.

步骤 2 检查 A_0 是否包含有向圈. 仍可能出现两种情况：

情况 1 A_0 没有有向圈，计算结束，A_0 就是 D_0 的以 v_1 为根的最小树形图.

情况 2 A_0 含有有向圈 $C_1, C_2, \cdots, C_s$，转入步骤 3.

步骤 3 把 D_0 中的 $C_1, C_2, \cdots, C_s$ 分别收缩成顶点 $u_1, u_2, \cdots, u_s$. D_0 中两个端点都属于同一个有向圈 C_i 的弧都被收缩掉了，其他的弧仍保留着，这样得到的图记作 D_1. D_1 中，

以收缩点为终点的弧 a_k 的长度要变化，变化的规律是：设 a_k 的终点是收缩点 v_i，并且有向圈 C_i 中一与 a_k 有相同终点的弧是 a_t，则

$$a_k \text{ 在 } D_1 \text{ 中的长度} = l(a_k) - l(a_t).$$

这里 $l(a_k)$ 与 $l(a_t)$ 分别表示弧 a_k 与 a_t 在 D_0 中的长度.

上面三个步骤的意义在上一节中都讲过了，现在再来继续研究，在得到 D_1 以后，下一步怎么办.

我们先讲一个定理，它说明了 D_0 与 D_1 的关系.

定理 7.1 设 D_1 是由 D_0 按上面的步骤 3 收缩有向圈 $C_1, C_2, \cdots, C_s$ 而得到的，则.

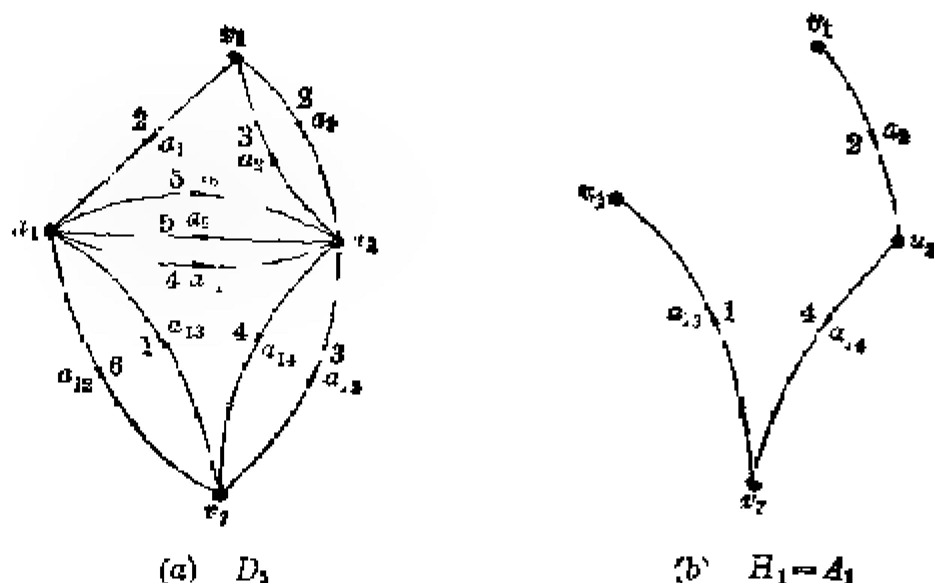
1. 如果 D_1 没有以 v_1 为根的支撑树形图，那末 D_0 也没有以 v_1 为根的支撑树形图.

2. 如果 H_1 是 D_1 的以 v_1 为根的最小树形图，那么按下述"展开"办法，得到的 H_0 就是 D_0 的以 v_1 为根的最小树形图. 展开办法是：

所有 D_1 中属于 H_1 的弧，在 D_0 中都仍属于 H_0；将每一个收缩点 v_i 展开成有向圈 C_i，C_i 中除去一条与 H_1 中的弧有相同终点的弧外，其他的弧都属于 H_0.

这个定理的证明这里就从略了.

我们现在通过例子把定理 7.1 中的展开办法再明确一下. 仍以图 7.5 为例，在得到 D_1 以后，对于 D_1 中不同于 v_1 的每一顶点，取一条以这个顶点为终点的最短弧，得到一个弧集合 A_1（见图 7.7(b)）. 显然 A_1 不包含有向圈，因此，按照上一节讲的，A_1 是 D_1 的最小树形图，也就是说 D_1 的最小树形图 H_1 已找到了，就是 A_1. 于是，按照定理 7.1，只要展开 H_1，就可以得到 D_0 的最小树形图 H_0 了，H_0 首先应该包含



(a) D_1　　　(b) $H_1=A_1$

图 7.7

H_1 中的弧 a_3, a_{13}, a_{14}. 另外,应该把 u_1, u_2 展开成 C_1, C_2,不过 C_1 中与 H_1 中的 a_{13} 有公共终点的 a_4 应该去掉, C_2 中与 H_1 中的 a_3 有公共终点的 a_8 应该去掉, 这样得到的 H_0 见图 7.8 (图中用虚线画的弧是展开时, 有向圈上去掉的弧, 它们不属于 H_0). 请大家考虑一下, 这种展开办法与上一节讲的"怎样修改 A_0 最合算"这一问题有什么关系?

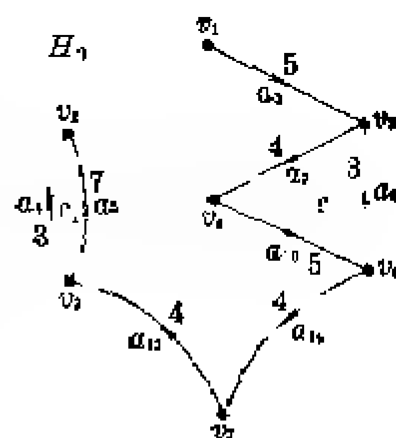
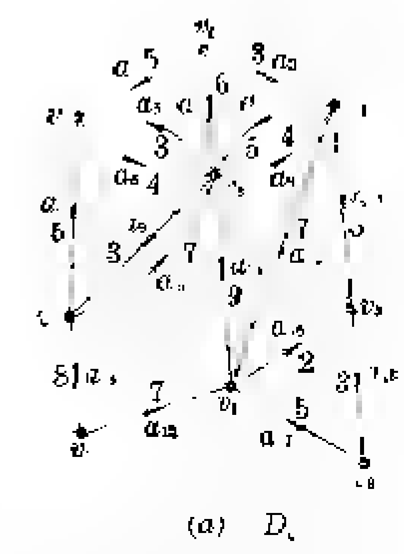
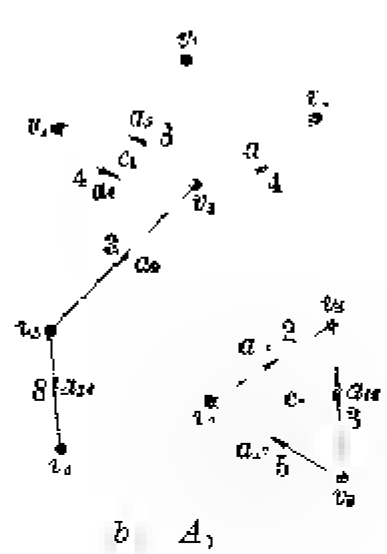


图 7.8

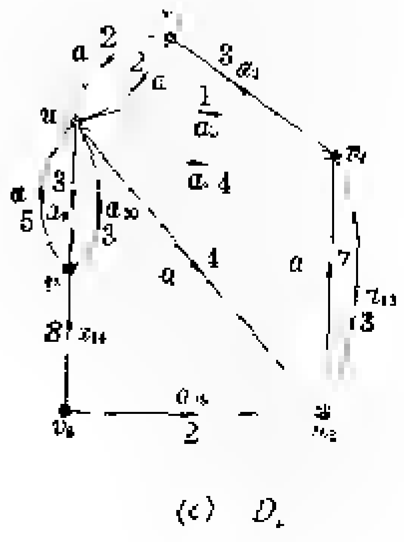
有了定理7.1, 再回过头来研究"在得到了 D_1 以后该怎么办"这一问题, 就很容易了. 答案应该是: 设法再求 D_1 的最小树形图, 求下来的结果如果是 D_1 没有树形图, 那末我们就可以根据定理7.1说 D_0 也没有树形图, 如果求得了 D_1 的最小树形图, 那末只要象刚才讲的那样一展开, 就得到 D_0 的最小树形图了. 也可以说, 定理7.1把求 D_0 的最小树形图问题归结为求 D_1 的最小树形图问题了.



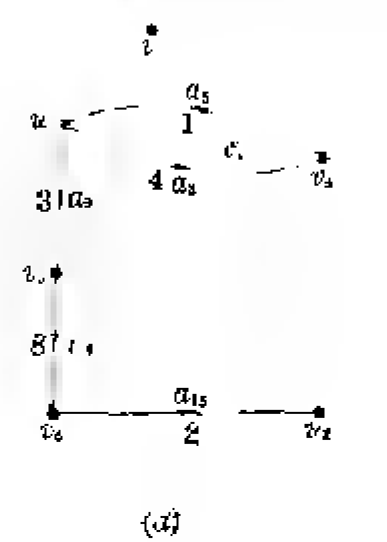
(a) D_1



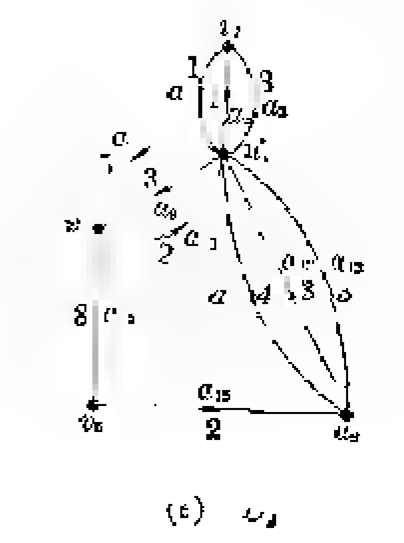
(b) A_1



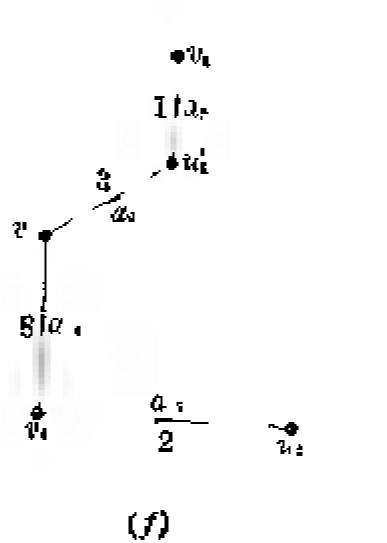
(c) D_2



(d)



(e) D_3



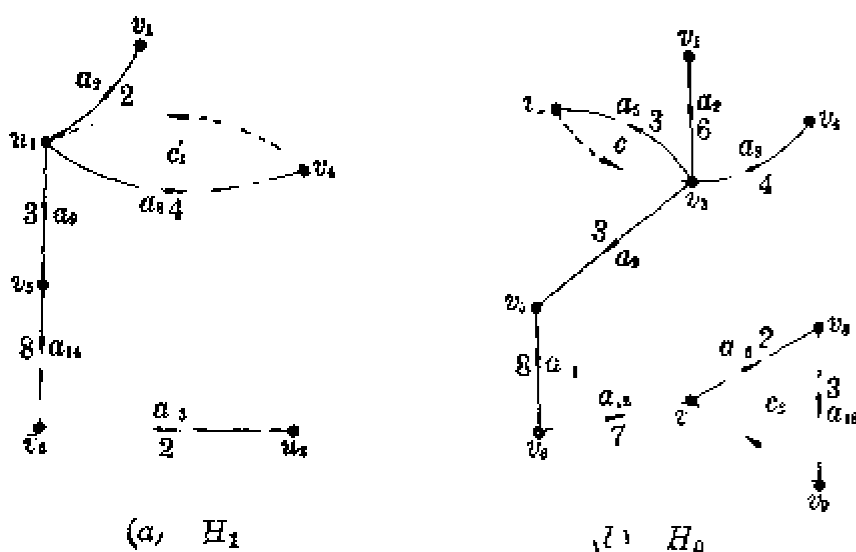
(f)

图 7 9

那末怎样求 D_1 的最小树形图呢？不用另找办法，只要再把上面讲的三个步骤用到 D_1 上去就行了，而且在进行步骤1（步骤1中得到的弧集合记作 A_1）时，对于不是收缩点的顶点 v_j 来说，以 v_j 为终点的最短弧不必求，因为 D_1 与 D_0 中以 v_j 为终点的最短弧显然是一样的，故实际上只要求出以收缩点为终点的最短弧，A_1 就得到了。

对于 D_1 来说，如果求得的 A_1 还包含有向圈，那末还要收缩，得到 D_2. 再把上述三个步骤用到 D_2 上去，求出 A_2，…. 这样做下去，一直到得到一个 D_s，它没有支撑树形图（这时可以肯定 D_{s-1}，…，D_1，D_0 也都没有支撑树形图），或它对应的 A_s 不包含有向圈，即 A_s 是 D_s 的最小树形图 H_s. 这时，通过展开求出 D_{s-1} 的最小树形图 H_{s-1}；再展开，……直到得到 D_0 的最小树形图 H_0 为止。

让我们再看一个例子. 对于图 7.9(a) 中的图 $D=D_0$，可以求得 (b) 中的 A_0，A_0 有两个有向圈. 收缩 D_0 得 D_1，再求出 A_1，A_1 有一个有向圈 c_1'，再收缩 D_1 得 D_2，现在求得的 A_2



(a) H_1　　(b) H_0

图 7.10

不包含有向圈了．因此 A_2 就是 D_2 的最小树形图 H_2，展开一次得 D_1 的最小树形图 H_1，再展开一次就得到 D_0 的最小树形图 H_0 了(见图 7.10)．

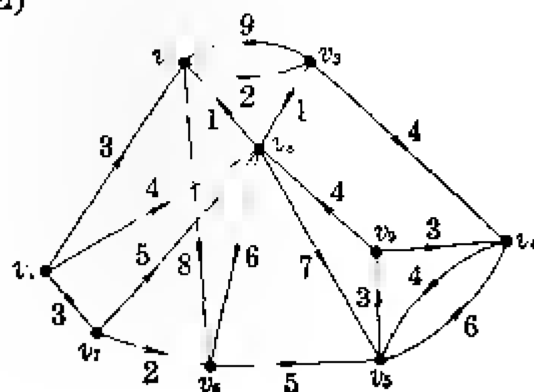
建议读者试试看，画一下这一节讲的计算方法的框图，并且证明这种计算方法是有效的．

习　题

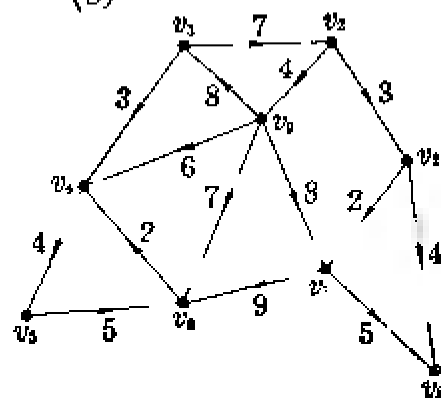
求下述有向图的以 v_1 为根的最 树形图。

(注意，答案可能是不存在支撑树形图)

(a)



(b)



八、最大流问题

1. 紧急运输问题

这一章要讲的最大流问题，也是一个很重要的极值问题．为了使大家能看清这个问题的实际意义，让我们从一个实际问题——紧急运输问题——讲起．

假设甲地急需大量某种物资，这种物资存放在仓库乙处，现在要把仓库乙的物资运到甲地去，需要运的物资很多，时间也很紧急，问应该怎样安排运输，才能使得在某一段时间内，运到甲地的物资最多．

例如我们把图 8.1 中的有向图 $G=(V, A)$ 看成一个公路网，v_1 是仓库乙所在地，即物资的起点，v_6 是甲地，即物资的终点．另外，每一条弧旁边都写了一个数．不过，与求最短路时不同了，这些数字在这里不再代表弧的长度，而是

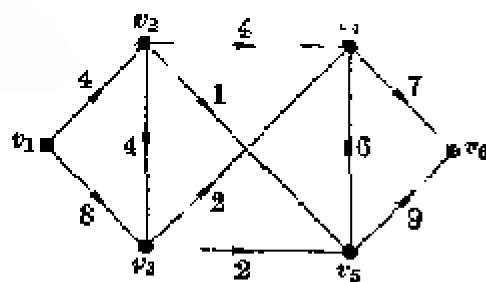


图 8.1

代表弧的“最大通过能力”(以后也叫“容量”)，就是在某一个时间里，可以通过这段公路的物资的最多数量．例如弧(1, 2)①

① 和前面一样，这一章中遇到的图都假设为简单图．因此我们可以用弧的起点和终点的号码来表示一条弧，(1, 2)就代表从 v_1 到 v_2 的弧．

的容量是 4，意思就是在某一时期内，在弧(1，2)上最多可以通过 4 百吨(图 8.1 中以百吨为单位)物资。

现在的问题就是：怎样安排运输，才能使得在某一个时期内从起点 v_1 运到终点 v_6 的物资最多？

一般地，在考虑运输问题时，往往首先考虑如何运输运费最节约，走的路程最短等等。但是在任务紧急时，考虑的着眼点就不放在省运费或省路程上，而在一段时间内尽量多运了。具有这种特点的运输问题就叫做紧急运输问题，在救灾、军火运输……等场合经常会遇到这类问题，另外，在电话、电报通讯中也会遇到类似的问题。

在紧急运输问题中，我们的目的是找一个最好的运输方案，也就是从起点运到终点的物资最多的运输方案。为此，我们首先要学会如何把一个方案表示出来。所谓一个运输方案，无非就是说：从起点运出多少物资，其中多少吨沿着这一条路运到终点，多少吨沿着那一条路运到终点……例如，对于图 8.1，我们就可以做出下面表 8.1 那样的一个方案：

表 8.1

有向路	经过哪些顶点	运输量
p_1	v_1, [illegible], v_4, v_6	2(百吨)
p_2	v_1, v_2, v_3, [illegible], [illegible]	2(百吨)
p_3	v_1, [illegible], v_5, v_6	1(百吨)
		[illegible](百吨)

但是，这里有一个问题，这个方案是不是行得通呢？因为

前面讲过，在我们考虑的这一时期内，能够通过每条弧的物资的数量不能超过这条弧的容量，超过了，这个方案就是行不通的。为了检查这个方案是不是行得通，我们可以把这个方案在图上表示出来，办法是：先在每条弧上都用深色写上数字。比如在有向路 p_1 经过的弧(1, 3)，(3, 4)，(4, 6)旁都填一个2。接着，再在 p_2、p_3 经过的弧上都填上2与1。同一个弧上如果填了几个数，中间就要插上"："号，[见图8.2(a)]，因为这些数字相加的结果，就是按照这个方案运输时，通过这条弧的物资的总数。在图8.2(b)中，已经把每条弧上的运输量都加起来了，有了图8.2(b)，就可以看出表8.1的方案确实是行得通的，因为每条弧旁的数字都没有超过弧的容量。

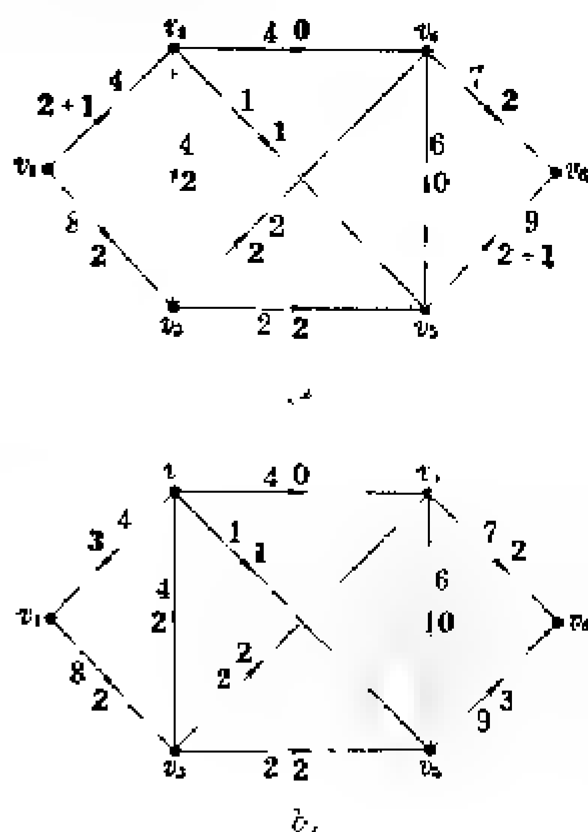


图 8.2

经过上面的分析，我们可以看出，单单用表8.1来表示一个运输方案是不够的，因为从表中看不出这个方案是不是行得通，还必须加上象图8.2(b)这样的一张图。这样看来，要表示一个运输方案还怪麻烦，要一张表加一张图才行。但是经过了一些分析研究，人们发现其实只要有象8.2(b)这样的一张图就够了。为什么呢？因为有了8.2(b)这样的一张图，就可以很容易地

"还原"出表 8.1. 不过，关于这个问题，这里就不仔细讲了，请大家先把图 8.2(b)"还原"一下试试，再做一下后面的习题 1，就清楚了.

今后，我们就用象图 8.2(b) 那样的图来表示一个方案. 仔细看一下图 8.2(b)，可以看出，最本质的东西实际上是那些黑体数字，或者说，我们实际上是用一组数来表示一个方案. 应该注意，并不是在每条弧边随便填一些数都能代表一个方案的. 这些数还必须满足一些条件，就是:

(1) 所填的数不能是负的;

(2) 所填的数不能大于这条弧的容量;

(3) 除了起点和终点以外，对于其他顶点 v_i 来说，所有指向 v_i 的弧上填的数的和应该等于所有从 v_i 出发的弧上填的数的和.

这三个条件中的前两个条件的必要性是显然的，条件(3)也是不难理解的，因为除了起点和终点以外，其余顶点都只是一些"中转点"，它们既不能输出物资也不应输入物资，因此运到它那儿的物资是多少吨，从它那儿运走的也应该是多少吨，不能多也不能少.

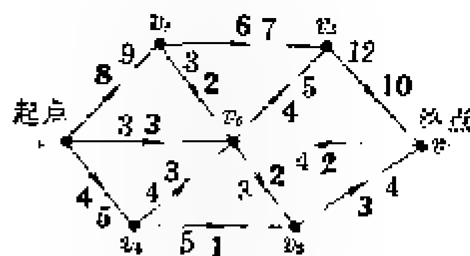
让我们来检验一下，图 8.2(b) 中的黑体数字是不是满足上面三个条件，一眼就能看出(1)、(2) 都成立. 对于条件(3)，我们应该分别对 v_2、v_3、v_4、v_5 进行检查. 例如对于 v_2，指向 v_2 的弧只有(1, 2)一条，通过它运到 v_2 的物资是 3 百吨，而从 v_2 出发的弧有(2, 3)，(2, 4)，(2, 5) 三条，通过它们从 v_2 运走的物资的数量是 $2+1+0=3$，也是 3 百吨. 因此对于 v_2 来说，条件(3)是成立的. 用同样的办法检查 v_3、v_4、v_5，可以发现对于它们来说，条件(3) 都成立.

让我们把上面讲的简单总结一下．前面这一段主要讨论表示一个运输方案的办法，结论是：可以用一组数(即填在弧旁的黑体数字)，表示一个方案，这些数必须满足前面讲的(1)(2)(3)三个条件．任意给出了这样的一组数，一定可以还原出象表 8.1 这样的一个方案．

对于一个方案来说，把从起点 v_1 出发的各条弧上的运输量加起来，得到的结果就是从 v_1 运出的物资的数量．而我们的目的就是从所有方案中，把能从 v_1 运出的物资最多的方案找出来，那末怎么找呢？关于这个问题，等我们在下一节中讲了网络、流、……一般概念后再介绍吧．

习　题

1. (a)检查一下右图中弧旁的黑体数字是否满足本节讲的条件(1)(2)(3)．



(b) 把右图"还原"成一个象表 8.1 那样的运输方案．能不能还原出几个不同的方案？

(c) 这个方案中，v_3 的运出量是多少？v_6 的运进量是多少？

2 证明：对于任意一个方案来说，起点的运出量一定等于终点的运进量．

2．网络上的最大流问题

上一节讲了紧急运输问题，这是一个实际问题．这一节再来提一个图论中的极值问题，就是网络上的最大流问题．大家一看就会明白，最大流问题就是从紧急运输问题(以及其他

类似的问题)中抽象出来的.

为了说明什么是最大流问题，必须先讲一下什么叫网络、什么叫流.

定义8.1 设$G=(V, A)$是一个简单有向图，它具有下面几个特点:

(1) 在G中指定了两个点，一个叫起点，一个叫终点. 象第三章中讲最短路时一样，我们就令v_1代表起点，v_n(n是V中顶点的个数)代表终点;

(2) G的每一条弧有一个非负的容量，设(i, j)是一条弧，那末(i, j)的容量就记作$c(i, j)$;

(3) G中不存在指向起点v_1的弧，也不存在从终点v_n出发的弧;

那么称G是一个网络.

很显然，图8.1中的图是一个网络.

定义8.2 设$G=(V, A)$是一个网络，又设它的每一条弧(i, j)对应有一个数$f(i, j)$，如果这些$f(i, j)$满足下面三个条件:

(1) 对于所有的弧(i, j)，有$f(i, j)\geqslant 0$;

(2) 对于所有的弧(i, j)，有$f(i, j)\leqslant c(i, j)$;

(3) 除了起点v_1和终点v_n外，对于其他顶点v_k来说，指向v_k的所有弧上的$f(i, k)$之和等于从v_k出发的所有弧上的$f(k, j)$之和;

那么称所有这些数$f(i, j)$的集合为网络G的一个流. 通常我们就用一个字母f代表一个流. 每一条弧(i, j)对应的$f(i, j)$叫做流f在弧(i, j)上的流量.

学了上一节的内容后，流的实际意义就很清楚了，

它就代表紧急运输问题中的一个运输方案.

以上一节的图 8.2 (b) 为例(重画在图 8.3 中), 令每条弧 (i, j) 旁边的黑体数字为 $f(i, j)$, 那末这些 $f(i, j)$ 就是一个流. 这里:

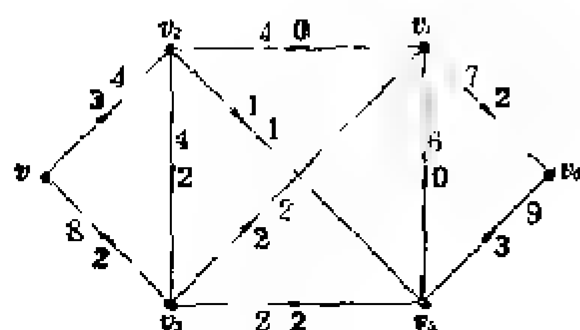


图 8.3

$c(1,2)=4$, $c(1,3)=8$,

$$c(2, 3)=4,\ c(2, 4)=4,\ c(2, 5)=1,\ c(3, 4)=2,$$

$$c(3, 5)=2,\ c(4, 6)=7,\ c(5, 4)=6,\ c(5, 6)=9;$$

而各条弧上的流量为:

$$f(1, 2)=3,\ f(1, 3)=2,\ f(2, 3)=2,\ f(2, 4)=1,$$

$$f(2, 5)=1,\ f(3, 4)=2,\ f(3, 5)=2,\ f(4, 6)=2,$$

$$f(5, 4)=0,\ f(5, 6)=3.$$

定义 8.3 设 f 是网络 G 的一个流. 则称从起点 v_1 出发的所有弧上的流量 $f(1, j)$ 之和为流 f 的值, G 的所有流中, 值最大的流叫做最大流.

定义 8.3 的意义也是很清楚的, 一个流 f 代表一个运输方案. 而 f 的值就是按这个方案运时, 从起点 v_1 运到终点 v_n 的物资的数量, 而最大流代表的就是从起点运向终点的物资最多的一个方案.

上面图 8.3 中的流的值是 $f(1, 2)+f(1, 3)=3+2=5$.

所谓网络上的最大流问题指的就是: 把一个网络 G 的最大流求出来. 显然, 这个问题会解了, 我们也就能为紧急运输问题找最好的运输方案了.

下一节就来介绍求一个给定网络的最大流的方法.

3. 最大流的求法

这一节开始讲求网络的最大流的方法．这种方法的轮廓是这样的：先作出一个流 f_1，然后检查 f_1 是不是最大流，如果是，很好，已经找到最大流了；如果不是，就设法把 f_1 改进成另一个流 f_2，而且 f_2 的值要比 f_1 的值大；然后再检查 f_2 是不是最大流，如果仍旧不是，就再把 f_2 改进成一个更好的流 f_3，……，这样做下去，直至找到一个最大流为止．

因此，这个方法由三个步骤组成．

步骤 1　求出第一个流．

步骤 2　检查现有的流是不是最大流，如果是，那末最大流已经找到了，计算结束；如果不是，转向步骤 3．

步骤 3　把现有的流改进成一个更好的流，转回步骤 2．

图 8.4 中画的就是上述算法的框图：

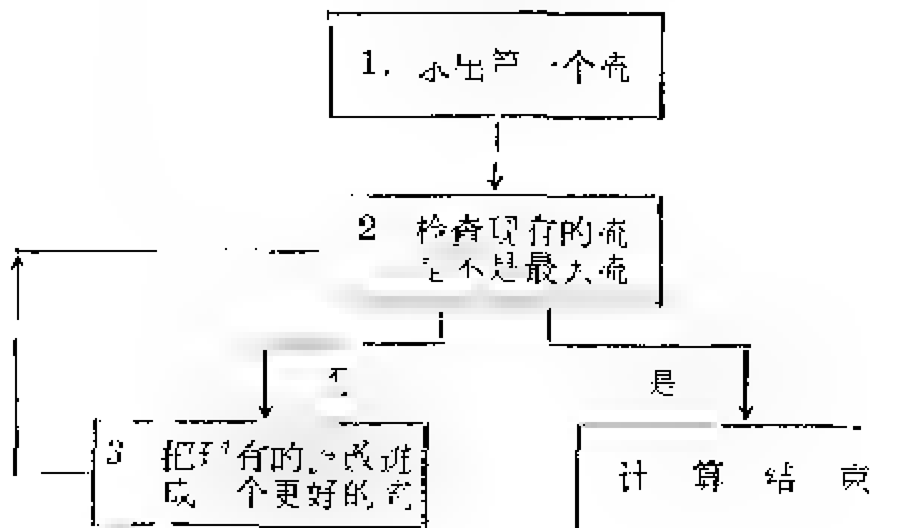


图　8.4

现在就来讲这三个步骤的具体做法．步骤 1 是很简单

的。因为只要令所有的 $f(i, j)$ 都等于零，就可以得到一个流。不难看出，定义 8.2 中的三个条件都是成立的。当然，取这个流作为第一个流并不理想，因为它的值是零，因此，如果我们有办法很容易地找出一个值大些的流 f_1，也可以取 f_1 作为第一个流。

下面我们把步骤 2 与 3 结合起来一起讲。先让我们分析一下，对于一个已知的流 f 来说，出现什么情况时，f 可以改进成一个更好的流。

首先，不难看出，在出现下述情况时，流 f 就可以改进

"存在一条从起点 v_1 到终点 v_n 的有向路 p，在 p 的每一条弧上，都有：$f(i, j) < c(i, j)$。"

理由很简单，因为我们可以让 p 的各条弧上的 $f(i, j)$ 都增加一个数 α（在紧急运输问题中，就相当于沿着 p 再多运 α 个单位的物资）。那末 α 应该等于几呢？让我们结合图 8.2(b) 中的流来讲（见图 8.5），在这里，有向路：

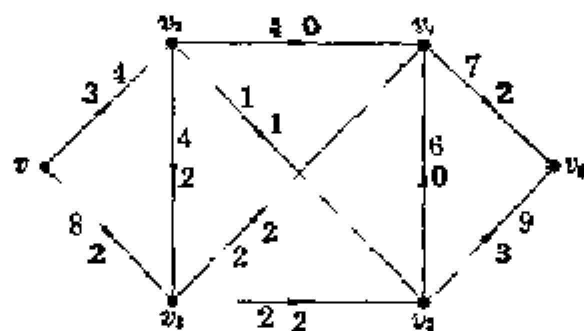


图 8.5

$$p = \{v_1, v_2, v_4, v_6\}$$

就具有上述性质，因为在它所包含的三条弧上，有：

$$f(1, 2)=3,\ f(2, 4)=0,\ f(4, 6)=2,$$

$$c(1, 2)=4,\ c(2, 4)=4,\ c(4, 6)=7,$$

$f(i, j)$ 都比 $c(i, j)$ 小。让这三个 $f(i, j)$ 都增加一个数 α，从弧 $(1, 2)$ 看，最多增加 $4-3=1$，从弧 $(2, 4)$ 看，最多增加

4 0 4, 而从弧 (4, 6) 看，最多增加 $7-2$ 5. 总起来说，最多能增加多少？想一想就会知道，只能增加1，因此 α 应该取1，也就是说，可以把图 8.5 中的流改进成图 8.6 中的流。经过这次改进，流的值增加了 α -1，即从5增加到了6.

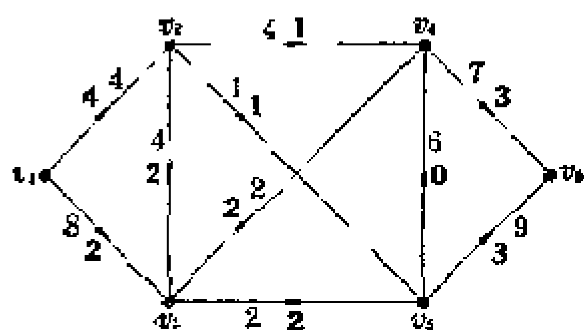


图 8.6

大家也许会问，一个流经过这样改进以后，得到的是不是仍旧是一个流，也就是说，定义 8.2 中的条件 (1)、(2)、(3) 是不是还都成立？答案是肯定的，不过这里就不写了，请大家自己证明一下吧！

请大家再试试看，在图 8.6 中，能不能再找到一条每段上流量＜容量的有向路。试一下就会知道，找不到了，那末是不是图 8.6 中的流就不能改进了呢？不是的，因为我们还有一种改进流的办法.

请大家看一下图 8.6 中的路：

$p=\{v_1, (1, 3), v_3, (2, 3), v_2, (2, 4), v_4, (4, 6), v_6\}$,

注意，p 是路而不是有向路(请复习一下第二章的定义 2.5，为了看起来清楚，我们把出现在 p 中的顶点和弧都写出来了)，沿着路 p 从起点 v_1 向终点 v_6 走时，弧 (1, 3)，(2, 4) 与 (4, 6) 的方向与前进方向一致，这样的弧叫做 p 的前向弧，

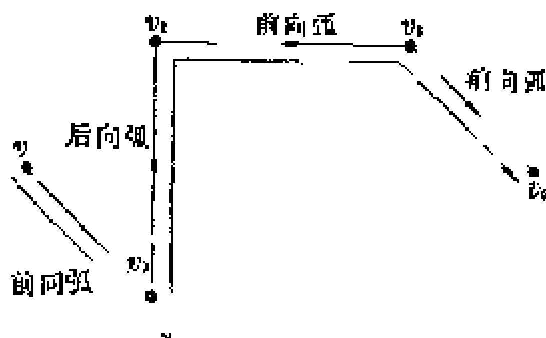


图 8.7

而弧(2, 3)的方向与前进方向相反，这样的弧叫做p的后向弧(见图8.7)。

定义8.4 设f是网络G的一个流，p是一条从起点v_1到某一顶点v_k的路p，它具有下述性质：

(1) 在p的所有前向弧上，$f(i, j)<c(i, j)$；

(2) 在p的所有后向弧上，$f(i, j)>0$；

那末称p为一条关于流f的可改进路。

不难看出，图8.7中的那条路p是关于图8.6中的流f的一条从v_1到v_6的可改进路。另外，前面讨论的从v_1到v_n的每条弧上都有$f(i, j)<c(i, j)$的有向路其实也是可改进路[因为路上没有后向弧，定义8.4中的(2)就不必考虑了]。

为什么我们把满足定义8.4的两个性质的路叫做"可改进路"呢？下面的定理就回答了这个问题。

定理8.1 设f是网络G的一个流，如果存在一条从v_1到v_n的关于f的可改进路p，那末f就可以改进成一个值更大的流f_1。

证明 这个定理证明的思路是：具体地给出一种方法，利用这种方法就可以把f改进成一个更好的流f_1，这种方法是：

(1) 不属于可改进路p的弧(i, j)上的流量一概不变，即对于这种弧来说，令$f_1(i, j)=f(i, j)$；

(2) 可改进路p上的所有弧(i, j)的流量都要变，其变化规则是：

在前向弧上，$f_1(i, j)=f(i, j)+a$，

在后向弧上，$f_1(i, j)=f(i, j)-a$。

α 叫做改进量，它应该按照下述原则来确定：α 既要取得尽量大，又要使变化后的 $f_1(i, j)$ 仍满足定义 8.2 中的条件 (1) 与 (2)，不难看出，按照这个原则，α 不能超过每条前向弧上的 $c(i, j)-f(i, j)$，也不能超过每条后向弧上的 $f(i, j)$，因此，α 应该等于前向弧上的 $c(i, j)-f(i, j)$ 与后向弧上的 $f(i, j)$ 中的最小值.

由上面讲的确定改进量 α 的办法立刻可以看出 f_1 满足定义 8.2 中的条件 (1) 与 (2)，另外也不难证明 f_1 满足 (3)（证明留给大家作为习题），因此 f_1 确实是一个流，并且它的值比 f 的值正好大 α，即它比 f 好．证完

让我们再回过头来看一下图 8.6 中的例子，前面我们已经指出了一条可改进路（见图 8.7）：

$p=\{v_1, (1, 3), v_3, (2, 3), v_2, (2, 4), v_4, (4, 6), v_6\}$

现在就按照定理 8.1 讲的办法来把图 8.6 中的流 f 改进成一个更好的流，首先应该定出改进量 α，先看 p 的前向弧，

在 (1, 3) 上，　$c(1, 3)-f(1, 3)=8-2=6$;

在 (2, 4) 上，　$c(2, 4)-f(2, 4)=4-1=3$;

在 (4, 6) 上，　$c(4, 6)-f(4, 6)=7-3=4$.

因此，总起来说，α 至多取 3．再看 p 的后向弧，这里只有 (2, 3) 一条，在这条弧上，$f(2, 3)=2$，因为后向弧上的流量要减少 α，为了保证减了以后不变负数，α 至多取 2，总起来说，α 应该取 2，这个改进过程见图 8.8．改进后的流的值为 8.

大家也许会问，定理 8.1 中讲的改进流的办法的实际意义是什么？看一下从图 8.8(a) 改进成 8.8(b) 的过程，这个问题就会清楚了．从图 8.8(a) 可以看出，弧 (1, 3) 上还有

潜力，但是由于从 v_3 出发的两条弧(3, 4), (3, 5)都已经“饱和”[即 $f(i, j)=c(i, j)$]，使得弧(1, 3)上不能再多运出物资来. 但是经过弧(3,4)(3, 5)的 4 百吨物资中，有 2 百吨是经过弧(2,3)运过来的. 于是，动一下脑筋就会发现，应该把弧(2, 3)上的 2 百吨物资退回到 v_2 去，改成从 (2, 4), (4, 6) 运，这样一来，弧(1, 3)上就又可以多运 2 百吨物资. 从而使得从 v_1 运出的物资的总量增加了. 定理 8.1 中讲的在 p 上改进的办法正是这种想法的反映.

(a)

(b)

图 8.8

定理 8.1 告诉了我们在什么情况下，一个流可以改进. 但是讲到现在，我们还一点也没有讲到怎样检查一个流是不是最大的. 下面的定理 8.2 就解决了这个问题.

定理 8.2 设 f 是网络 G 的一个流，如果不存在从 v_1 到 v_n 的关于 f 的可改进路 p，那末 f 一定是最大流.

这个定理的证明比较长，就不讲了. 大家不要以为这个定理可以用以下的简单办法来证明："既然找不到可改进路 p，这个流就不能改进，当然就是最大流了."这种证法是不对的，因为虽然找不到可改进路，很可能还有其他的办法来改进

现有的流.

有了定理 8.2 以后，前面讲的求最大流的三个步骤就更具体了，那就是：

步骤 1　用任何方便的办法求出第一个流，或者取所有 $f(i, j)=0$ 作为第一个流；

步骤 2　对于现有的流 f，找找看是否存在可改进路，如果不存在，现有的流就是最大流，如果找到了可改进路，转向步骤 3；

步骤 3　按照定理 8.1 中讲的方法把现有的流改进成一个更好的流，转向步骤 2.

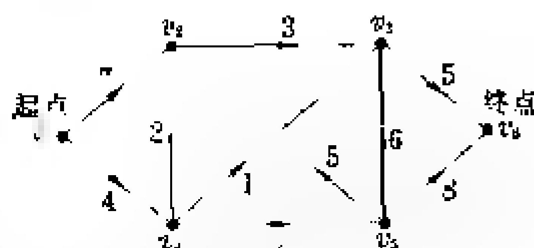
从图 8.5 中的流改进成图 8.6 中的流，接着又改进成图 8.8 中的流其实就是按照这个步骤在做. 不难看出，对于图 8.8 的流来说，不可能再找到可改进路了(请大家试试看)，因此由定理 8.2，可以肯定图 8.8 中的流已经是一个最大流了.

现在来研究一下，用刚才讲的步骤 1，2，3 来求最大流还有什么缺点. 步骤 1 和 3 做起来很简单，但是步骤 2 就不行了. 大家可以想象，如果给你一个比较大的网络，比如说有几百个点，几百条弧，要检查一个已知的流是不是最大流，就是要找找有没有可改进路. 假如你已经费了两个小时，还没有找到，你是不是就可以说找不到了呢？ 当然不能. 因此，我们必须要有一种系统地寻找可改进路的方法，这种方法必须具有下述特点：当存在可改进路时，能很快的把它找出来，当不存在可改进路时，能很快地告诉你："不用再找了，肯定不会有了." 下一节讲的标号法就是这样的一种方法.

习　题

1. 证明：用定理 8.1 中讲的方法来改进一个流，改进后得到的一定仍是一个流。

2. 为右图的网络找一个最大流。



3. 设一个网络的每一条弧的容量都是整数，试证明一定存在一个最大流，它在每条弧上的流量都是整数。[提示：只要取所有 $f(i, j)=0$ 作为第一个流，然后证明每次按照定理 8.1 的办法改进流时，改进量 θ 一定是整数。]

4．找可改进路的方法——标号法

这一节要介绍的找可改进路的方法也叫标号法，当然具体内容和第三章中求最短路的标号法是不同的。首先明确一下，一个顶点 v_i 在什么情况下能得到标号？规定是：如果你能够肯定存在着从起点 v_1 到 v_i 的可改进路，顶点 v_i 就应该得到标号。因此，一旦 v_n 得到标号，就可以肯定存在 v_1 到 v_n 的可改进路了。

让我们结合具体例子来讲标号的办法，仍以图 8.6 中的流为例，开始时令 v_1 是已标号点，并且给 v_1 以标号 $(0, +)$。[这个标号 $(0, +)$ 没有什么意义，只不过说明 v_1 已经有了标号罢了]，然后检查一下从 v_1 出发的弧，这里有 $(1,2)$，$(1,3)$ 2 条，先看 $(1, 3)$，因为 $f(1, 3)=2<c(1, 3)=8$，因此单单一条弧 $(1, 3)$ 就构成一条从 v_1 到 v_3 的可改进路，也就是说，可以肯定存在从 v_1 到 v_3 的可改进路了，因此 v_3 应该得到标号，

我们给 v_3 以标号 (1, +), 这里 (1, +) 的意思是: v_3 的标号是由一条从 v_1 出发("+"代表出发)的弧, 即 (1, 3), 得来的. 再看弧 (1, 2), 因为 $f(1, 2)=c(1, 2)=4$, 不能给 v_2 标号. 在对 v_1 出发的弧都进行了检查以后, 我们在 v_1 的标号 (0, +) 下面划一条线, 表示对 v_1 已经检查过了, 以后就不必再考虑它了(见图 8.9).

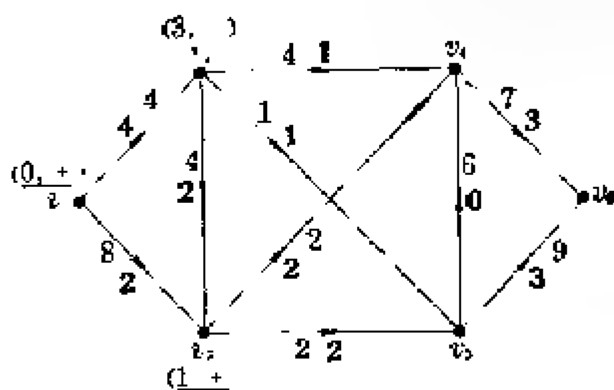


图 8.9

一般情况下, 我们可以按下面的办法来进一步扩大已标号点的范围: 设 v_i 是已标号点, 那末应该存在从 v_1 到 v_i 的可改进路. 这时, 如果存在一条以 v_i 为起点的弧 (i, l), 满足 $f(i, l)<c(i, l)$, 那末显然可以把从 v_1 到 v_i 的可改进路延长一条弧, 使它成为一条从 v_1 到 v_l 的可改进路. 因此, 如果 v_l 还没有得到标号, 就可以给 v_l 标号 $(i, +)$, 并且把 v_l 改为已标号点(见图 8.10). 同样, 如果存在一条以 v_i 为终点的弧 (k, i) 满足 $f(k, i)>0$, 并且 v_k 还没有得到标号, 则给 v_k 以标号 $(i, -)$ 并且把 v_k 改为已标号点. 这里标号 $(i, -)$ 的意思是 v_k 的标号是从一

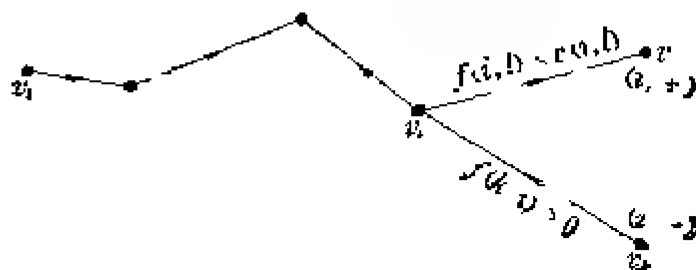


图 8.10

条指向（"→"表示指向）v_i的弧得来的，也就是从(k, i)得来的．

回到图8.9，现在有两个已标号点，v_1已检查过了，不必再考虑，因此只考虑v_3，现在来看看能不能用上面讲的办法扩大标号点的范围，从v_3出发的弧有$(3, 4)$和$(3, 5)$2条，这2条弧上都有$f(i, j)=c(i, j)$，因此不能对v_4与v_5进行标号，再看指向v_3的弧，这里有$(1, 3)$，$(2, 3)$2条，因为$f(2, 3)>0$，所以可以给v_2以标号$(3, -)$，在弧$(1, 3)$上，虽然也有$f(1, 3)>0$，但是v_1已经有标号了，所以不必对v_1重新标号．把从v_3出发和指向v_3的弧都考察过后，在v_3的标号下面也划一条线，表示对v_3已检查过了

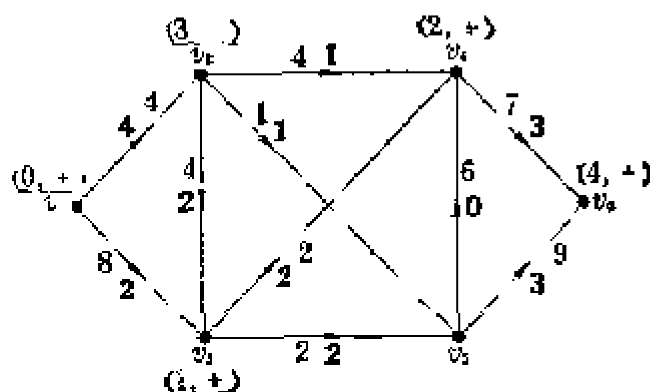


图 8.11

（见图8.9）．接着，再考虑已标号而未检查的点v_2．因为在弧$(2, 4)$上，$f(2, 4)<c(2, 4)$，所以v_4可以得到标号$(2, +)$，其他顶点都不能从v_2得到标号．再检查v_4，发现v_6可以得到标号$(4, +)$，v_6一得到标号，标号过程就可以停止了，因为我们可以肯定存在从v_1到v_6的可改进路了．

接下去，还应该把从v_1到v_6的可改进路p找出来，找p的方法叫做"倒向追踪"，就是从v_6开始利用顶点上的标号倒回去找，v_6的标号是$(4, +)$，说明"v_6是从一条v_4出发的弧，即$(4, 6)$，得到标号的"，因此，p的最后一段应该是：

$$\{v_4, (4, 6), v_6\},$$

因为 v_4 的标号是 (2, +)，所以上面的路应扩大成：

$$\{v_2, (2, 4), v_4, (4, 6), v_6\},$$

注意，v_2 的标号是 (3, -)，所以 v_2 前面的弧应该是 (2, 3) [而不是 (3, 2)]，即上面的路应该扩大成：

$$\{v_3, (2, 3), v_2, (2, 4), v_4, (4, 6), v_6\},$$

最后，因为 v_3 的标号是 (1, +)，所以上述路又可以扩大成：

$$p=\{v_1, (1, 3), v_3, (2, 3), v_2, (2, 4), v_4, (4, 6), v_6\},$$

已经出现了 v_1，不必再往下做了，因为需要找的从 v_1 到 v_n 的可改进路已经找到了。在找到了可改进路以后，就用定理 8.1 的方法把现有的流改进成一个更好的流。注意，在对新的流再找可改进路时，原有的标号应全部去掉，再从 v_1 开始重新标号。

用上面讲的办法来找可改进路，计算起来是很快的，因为每一个检查过的顶点以后就不用再考虑了，也就是说，一个顶点至多考虑一次。另外，不难看出，如果存在从 v_1 到 v_n 的可改进路，那末，v_n 一定会得到标号的。为什么呢？让我们结合图 8.12 中画的一条可改进路的示意图来讲。首先 v_1 一开始就是已标号点，在考虑 v_1 时 v_2 就会得到标号。再看 v_3，我们来证明 v_3 一定会得到标号。可以这样想，当考虑顶点 v_2 时，可能出现两种情况：一是 v_3 已经得到标号了，那末很好，已经证明 v_3 得到标号了。另一种情况是 v_3 还没有得到标号，



图 8.12

那末这时按标号的办法，就会给 v_5 以标号 (2, －)．这就证明了不管出现什么情况，v_3 一定会得到标号，继续推下去，就可以知道 v_4、…、v_n 也都一定会得到标号．

既然在存在从 v_1 到 v_n 的可改进路时，v_n 一定会得到标号．那末如果出现 v_n 还没有得到标号而又无法继续扩大已标号点的范围时，说明什么问题呢？大家想一想就会知道，这种情况就说明了对于现有的流来说，不存在从 v_1 到 v_n 的可改进路，因此根据定理 8.2，就可以肯定现有的流已经是最大流了．

让我们再来看一个例子，就以图 8.8(b) 中的流为例(见图 8.13)，首先给 v_1 标号 (0, ＋)，接着 v_3 可以得到标号(1, ＋)再考虑 v_3，不管是从 v_3 出发的弧，还是指向 v_3 的弧，都不能使任何顶点得到标号，检查完了 v_3 以后，发现不存在已标号而未检查的顶点了，因此也就无法再扩大已标号点的范围了，但是这时终点 v_6 还没有得到标号．因此，按照前面讲的，马上可以肯定不存在可改进路，即这个流已经是最大流了．

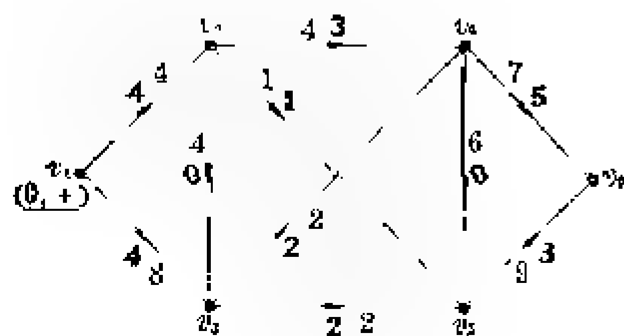
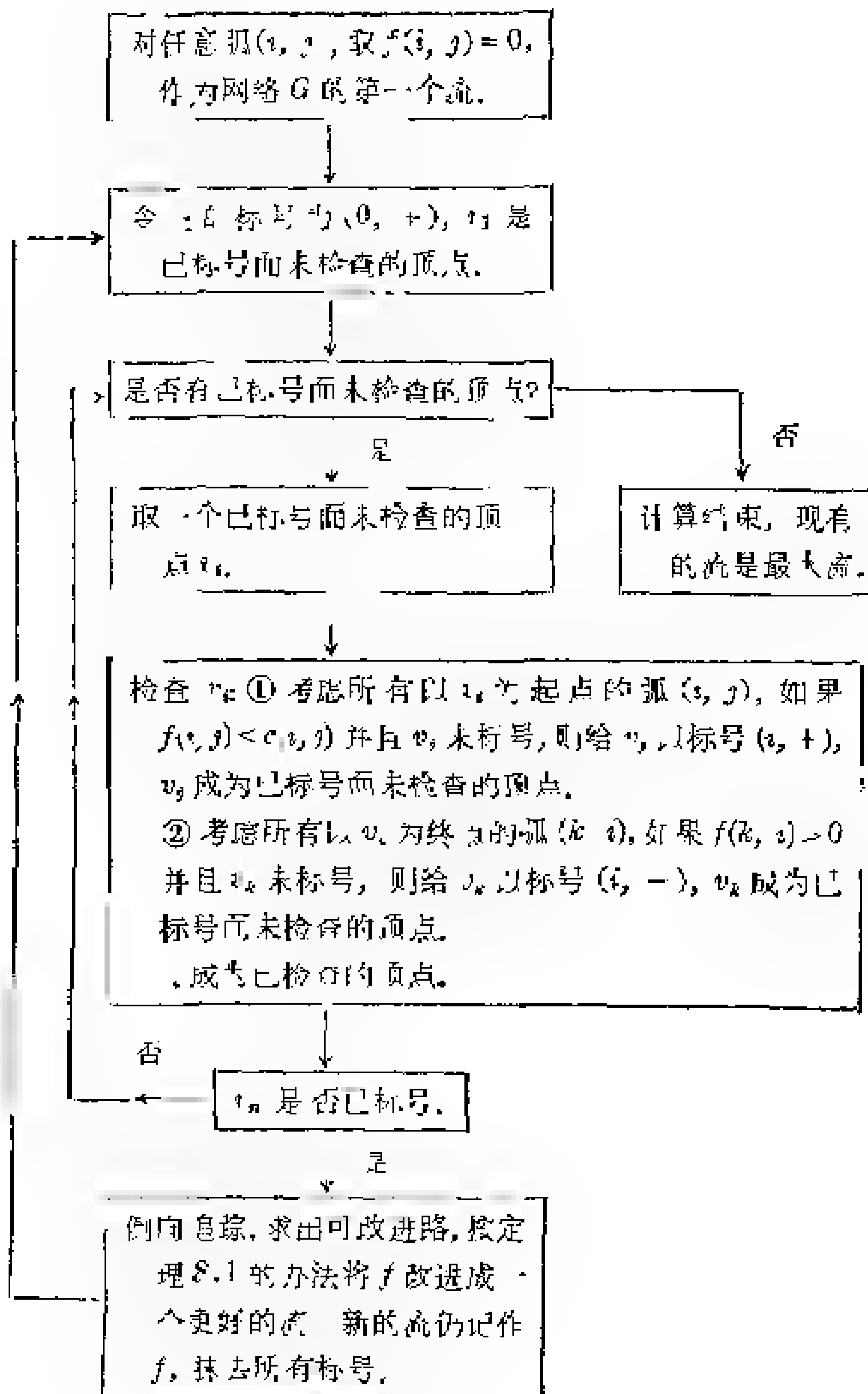


图 8.13

讲到这里，求网络最大流的计算方法已经讲完了，下面给出的求网络最大流的计算方法的框图是上一节和这一节讲的内容的总结，请大家结合着这个框图把整个方法再回想一遍．



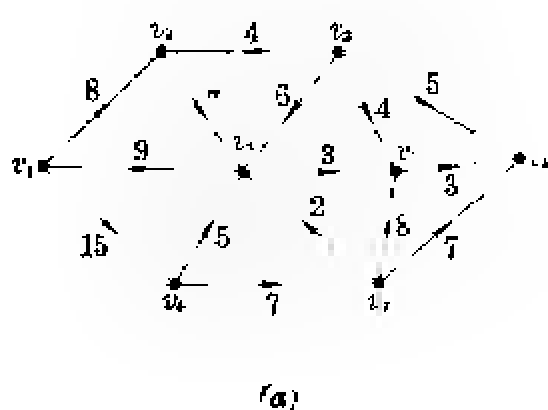
对任意弧(i, j)，取$f(i, j)=0$，作为网络G的第一个流.
令v_1的标号为$(0, +)$，v_1是已标号而未检查的顶点.
是否有已标号而未检查的顶点?
是
否
取一个已标号而未检查的顶点v_i.
计算结束，现有的流是最大流.
检查v_i：① 考虑所有以v_i为起点的弧(i, j)，如果$f(i, j)<c(i, j)$并且v_j未标号，则给v_j以标号$(i, +)$，v_j成为已标号而未检查的顶点.
② 考虑所有以v_i为终点的弧(k, i)，如果$f(k, i)>0$并且v_k未标号，则给v_k以标号$(i, -)$，v_k成为已标号而未检查的顶点.
v_i成为已检查的顶点.
否
v_n是否已标号.
是
倒向追踪，求出可改进路，按定理8.1的办法将f改进成一个更好的流，新的流仍记作f，抹去所有标号.

图 8.14

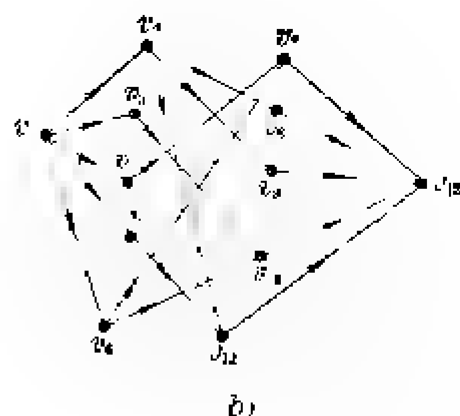
最后，再来讨论一下上面讲的计算方法的有效性问题．要估计这种方法的运算次数是比较难的，关键问题是不容易估计需要改进几次才能得到最大流．因为改进的次数不仅与网络中的顶点个数有关，还和弧的容量以及标号时的具体做法有关．有时会出现这样的情况，一个网络的顶点倒不多，但是有些弧的容量是很大的数，结果需要改进很多次（改进的次数不是顶点个数 n 的多项式）才能得到最大流．不过，已经有人指出，只要把我们讲的方法稍为修改一下，就可以得到一种有效的计算方法．关于这个问题，这里就不仔细讲了（有兴趣的读者可以参阅更专门的书籍），大家只要记住这样的结论：对于网络最大流问题来说，是存在有效的计算方法的．

习　　题

找出下面两个网络的最大流．



(a)



(b)

（所有的弧的容量都假设为1）

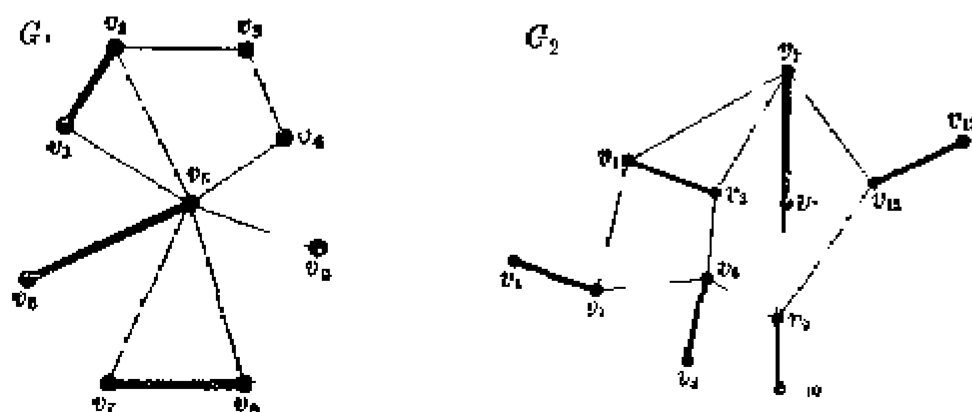
九、二分图的最大匹配问题

1. 匹配、二分图

在这本书的引言中，我们曾经讲过一个飞机驾驶员的搭配问题，那里也介绍了“匹配”这一概念，这一章与下一章就专门研究与匹配有关的一个极值问题．首先让我们把匹配的定义再明确地讲一下．

定义 9.1 设 $G=[V, E]$ 是一个无向图，$M\subseteq E$ 是 G 的若干条边的集合，如果 M 中的任意两条边都没有公共的端点，就称 M 是一个匹配．

例如图 9.1(*a*)、(*b*)中的粗边组成的集合 M_1 与 M_2 分别是图 G_1 与 G_2 的匹配．



(*a*) 图 G_1 中粗边的集合 M_1　　(*b*) 图 G_2 中粗边的集合 M_2

图 9.1

应该注意，对于任意图 G 来说，如果取 $M=\phi$，则 M 也是 G 的一个匹配(请想一下，定义 9.1 中的条件是否成立?).

有好几个极值问题与匹配这一概念有关. 我们要研究的问题是，从给定的图 $G=[V, E]$ 的所有匹配中，把包含边数最多的匹配找出来.

一般，我们把边数最多的匹配叫做**最大匹配**. 因此上面讲的问题就叫做**最大匹配问题**. 在这本书的引言中已经说过，飞机驾驶员的搭配问题就可以归结为最大匹配问题.

一般说来，要把一个图 G 的最大匹配找出来，不是一件很容易的事，不过，对于一种叫做二分图的特殊的图来说，最大匹配问题却比较容易解决. 因此，在这一章中，我们首先来讲一下二分图的最大匹配的求法，下一章再研究一般图的情况.

首先，让我们讲一下什么叫二分图.

定义 9.2 设 $G=[V, E]$ 是一个无向图，如果可以把 G 的顶点集合 V 分成两个部分 X 与 Y，使得 E 中的每一条边的两个端点，一定是一个属于 X 而另一个属于 Y，就称 G 是一个二分图. (为明确起见，有时把二分图记作 $G=[X, Y; E]$.)

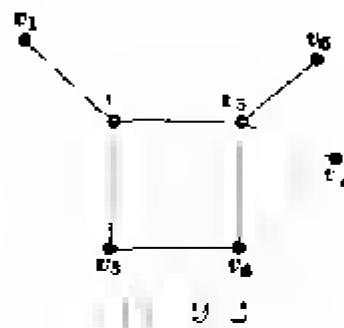


图 9.2

例如图 9.2 中的图 G，就是一个二分图. 因为我们可以令:

$$X=\{v_1, v_3, v_5\},$$

$$Y=\{v_2, v_4, v_6, v_7\}.$$

不难检查，每条边的两个端点都是一个属于 X 而另一个属于 Y.

通常，在画一个二分图时，我们把属于 X 的顶点画成一

行，把属于 Y 的顶点画成另外一行，并且把属于 X 与 Y 的顶点分别记作 $x_1, x_2, \cdots, x_r$ 与 $y_1, y_2, \cdots, y_t$，即画成图9.3那样。由二分图的定义可知，不存在连接 X 中的任两个顶点的边，也不存在连接 Y 中的任两个顶点的边。

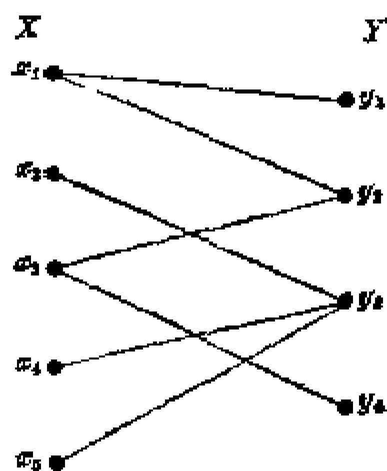
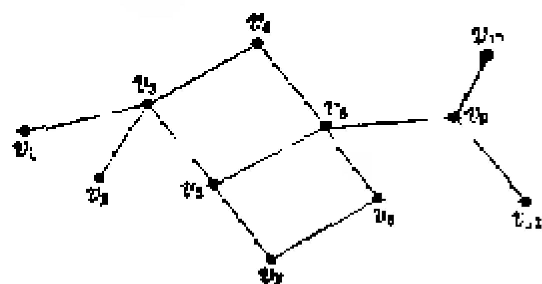


图 9.3

求二分图的最大匹配问题也有一定的实际意义。例如一个空军部队中，有一部分正驾驶员、一部分副驾驶员，而每架飞机必须有一个正驾驶员和一个副驾驶员。不难看出，"如何搭配正副驾驶员才能使出航的飞机最多"这样的问题，可以归结为一个二分图上的最大匹配问题。

习　　题



1. 判断一下，左面的图 G 是不是二分图。如果是的话，请说明 X 与 Y 应分别包含哪几个顶点。

2. 用直接观察的办法给图9.3中的二分图找一个最大匹配。

2. 二分图的最大匹配的求法

现在就来讨论求二分图的最大匹配的一种方法。这种方法是以上一章讲的求网络的最大流的方法为基础的。

我们就结合图9.3中的二分图 G 来讲，从图 G 出发，可

以造一个网络 G，$\bar{G}$ 是按照下述办法造出来的：

(1) 增加两个顶点 s 与 t，s 为网络的起点，t 是终点.

(2) 从 s 向 Y 的每一个顶点都画一条有向弧，从 Y 的每一个顶点都向 t 画一条有向弧.

(3) 原来 G 中的边都改为有向弧，方向都是从 X 的顶点指向 Y 的顶点.

(4) 令所有弧的容量都等于 1.

图 9.4 中画的就是二分图 G 及由它出发造出来的网络 $\bar{G}$.

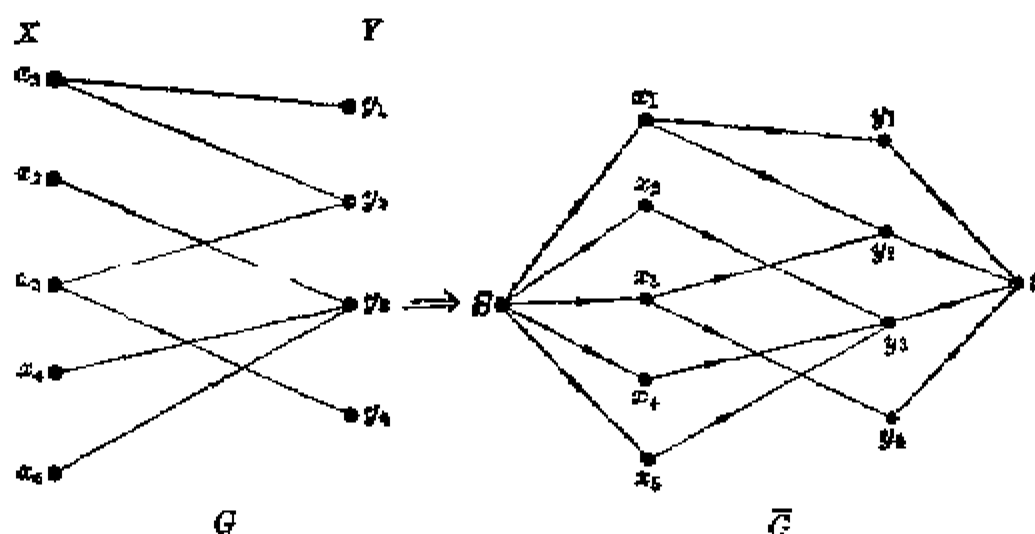


图 9.4

现在来分析一下 $\bar{G}$ 与 G 之间有什么关系，首先，我们有：

引理 9.1 如果 G 有一个由 p 条边组成的匹配 M，那末 $\bar{G}$ 就有一个流 f，它的值是 p.

这个引理的证明是很显然的，大家只要看一下图 9.5 就会明白了. 图 9.5(a) 中的粗线表示 G 的一个由 2 条边组成的匹配，而 (b) 中画的就是与它对应的一个值是 2 的流 f（如果一条弧的旁边没有黑体字，就表示流 f 在这条弧上的流量

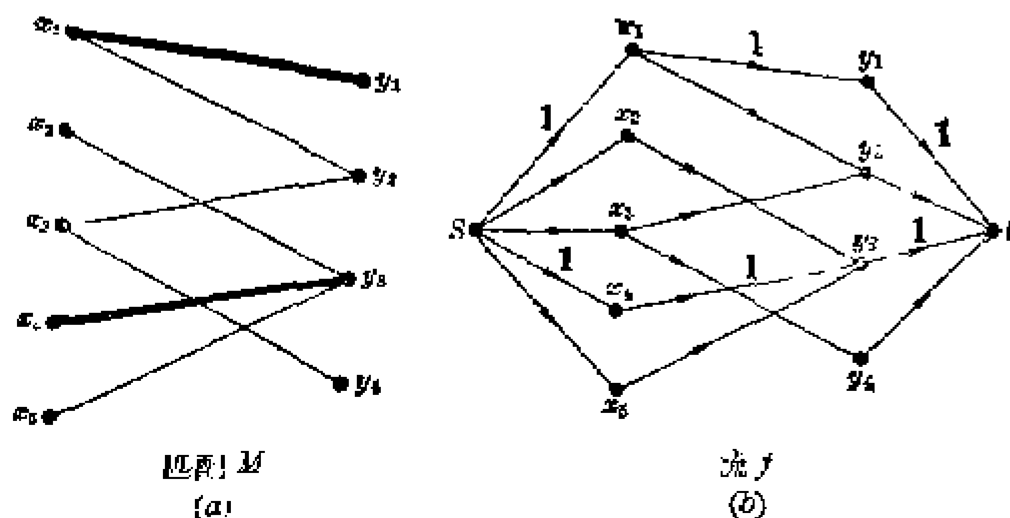


图 9.5

是 0).

从引理 9.1 还可以得出一个结论，就是：

引理 9.2 设 G 的最大匹配包含 m 条边，$\bar{G}$ 的最大流的值是 v，那末：

$$v \geqslant m$$

证明 由引理 9.1，$\bar{G}$ 一定有一个值为 m 的流，当然 $\bar{G}$ 的最大流的值 v 就一定大于或等于 m 了．证完．

进一步考虑一下，反过来，如果先有了 $\bar{G}$ 的一个流 f，它的值是 p，是不是一定能找到 G 的一个包含 p 条边的匹配呢？这个问题稍为复杂一些了，为什么呢？因为 $\bar{G}$ 很可能会有流量不是整数的流，例如图 9.6 中就画了 $\bar{G}$ 的一个流，它的值虽然是整数 2，但是却有好几条弧上的流量都不是整数．但是，如果我们加一个条件，就是：如果我们只考虑 $\bar{G}$ 的一种特殊的流，即值都是整数，

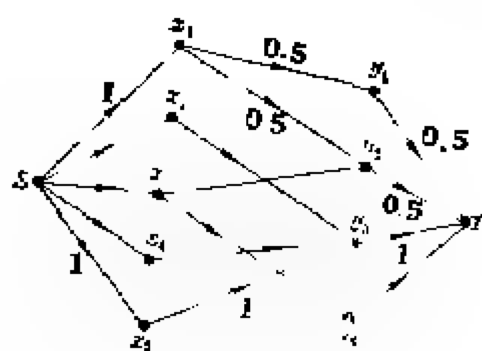


图 9.6

并且每一条弧上的流量都是整数的流，那时就有：

引理9.3 如果 $\bar{G}$ 有一个值为整数 p 并且每条弧上的流量也都是整数的流 f，那末 G 就有一个由 p 条弧组成的匹配。

证明 因为 $\bar{G}$ 内每一条弧的容量都是1，而流 f 在每一条弧上的流量都是整数，很显然，这些流量不是0就是1。

不难看出，在这种情况下，一定有：

(1) 恰好有 p 条从 s 指向 X 的顶点的弧，这些弧上的流量是1。

(2) 恰好有 p 条从 Y 的顶点指向 t 的弧，弧上的流量是1。

(3) 恰好有 p 条从 X 的顶点指向 Y 的顶点的弧，弧上的流量是1，而且这 p 条弧中，既不可能有两条弧有共同的起点，也不可能有两条弧有共同的终点。

因此，(3)中的 p 条弧去掉箭头后，就恰好是 G 中的一个由 p 条边组成的匹配。证完。

为了把引理9.3的证明弄清楚，请大家结合图9.5再考虑一下，请大家给 $\bar{G}$ 任意找几个流量都是整数的流，看看是不是一定具有上面讲的三个特点。

有了上面的准备，再来证明下面的定理就不难了。

定理9.1 设 G 的最大匹配包含 m 条边，$\bar{G}$ 的最大流的值是 v，那末

$$v = m$$

证明 由于 $\bar{G}$ 的每条弧上的容量都是整数，按照上一章所讲的(见第八章第3小节习题3)，一定可以找到一个各条弧上的流量都是整数的最大流，当然它的值也是 v，于是由引理9.3可知，G 中一定存在一个匹配，包含 v 条边，这就证明了

G 的最大匹配所包含的边数 $m\geqslant v$. 另一方面，在引理 9.2 中，我们已经证明了 $v\geqslant m$，$\therefore v=m$. 证完.

上面几个引理与定理的证明过程中，其实已经包含了求一个二分图 G 的最大匹配的一种方法，就是：

第一步：从 G 出发造出网络 $\bar{G}$；

第二步：求出 $\bar{G}$ 的一个值为 v 的最大流，它在每条弧上的流量都是整数；

第三步：按引理 9.3 证明过程中所讲的那样，找出 G 的一个由 v 条边组成的匹配 M，那末 M 就是要求的 G 的最大匹配了.

现在就用上面的方法来求图 9.4 中的二分图 G 的最大匹配. $\bar{G}$ 也已经画在图 9.4 上了. 接下来求 $\bar{G}$ 的最大流. 我们从所有 $f(i, j)=0$ 开始，经过三次改进，得到图 9.7 中的值为 3 的流，把上一章讲的标号法用到图 9.7 上，易见对于这个流来说，找不到可改进路了(请大家自己计算一下!). 因此，这个流是最大流，并且显然它在每条弧上的流量是整数，由这个流可以得到原来图 G 的一个由三条边组成的匹配，即图 9.8 中的粗线组成的匹配.

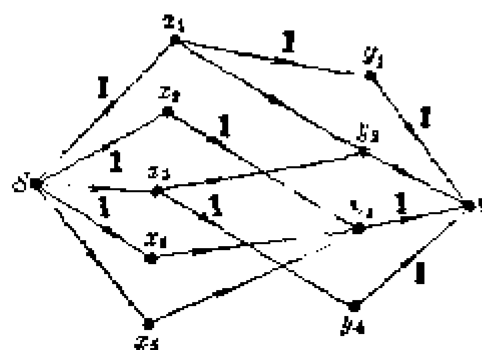


图 9.7

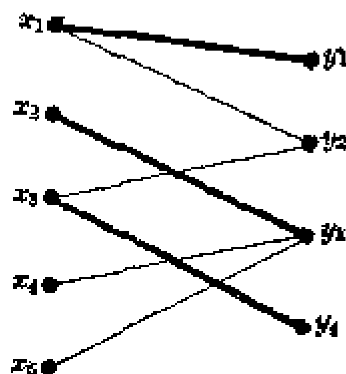
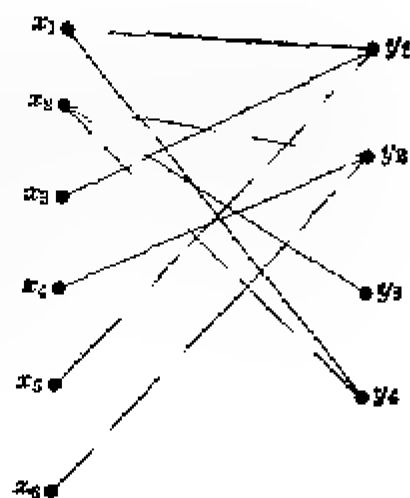


图 9.8

习 题

1 求出右面二分图的最大匹配.



2 某空军小队有正、副驾驶员各五人，分别记作 A_1, A_2, …, A_5 与 B_1, B_2, …, B_5，下面的表说明了每一个正驾驶员能与哪几个副驾驶员同机飞行（✓ 表示可以同机飞行，× 表示不能同机飞行），现设每架飞机必须有正、副驾驶员各一人，请寻找一个出航的飞机最多的方案.

副 / 正	B_1	B_2	B_3	B_4	B_5
A_1	×	✓	×	✓	×
A_2	✓	×	✓	×	×
A_3	×	✓	×	✓	×
A_4	×	×	✓	×	✓
A_5	×	✓	×	✓	×

十、任意图的最大匹配问题

1、交错路、可扩充路

上一章我们利用了求网络最大流的方法，顺利地解决了求二分图的最大匹配问题．当然，接下去，该考虑怎样求一个一般图 G（即 G 不一定是二分图）的最大匹配了．也许，大家马上又会想到，能不能把这个问题也归结成网络问题来解决？有不少人试验过，但是没有成功．因此要另找办法．经过一些数学家的研究，发现要解决最大匹配问题，必须先找出一些研究匹配理论的"工具"．下面我们将要介绍的交错路、可扩充路等概念，就是这样的一些工具．

下面几个定义，都是相对于图 G 的某一个给定的匹配 M 来说的．

定义 10.1 设 v_i 是图 G 的一个顶点，如果 v_i 不与任意一条属于匹配 M 的边关联，就称 v_i 是一个未盖点．

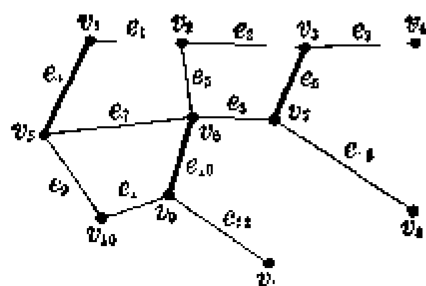


图 10.1

令图 10.1 中的图 G 与由粗线组成的匹配 M 来说，v_2, v_4, v_8, … 都是未盖点．

为什么采取未盖点这样一个名称呢？直观地说，在给定了一个匹配 M 以后，我们可以认

为每一条属于 M 的边 e_i“盖住”了两个顶点，就是 e_i 的两个端点．例如图 10.1 中的 e_4 盖住了 v_1 与 v_5，e_6 盖住了 v_3 与 v_7，等等，而其余的顶点，也就是没有被盖住的顶点，自然就叫做未盖点了．

定义 10.2 设 p 是图 G 的一条初级路（就是顶点不重复出现的路），如果 p 的任意两条相邻的边一定是一条属于 M 而另一条不属于 M，就称 p 是一条交错路．

图 10.2 中画的是从图 10.1 的图 G 中找出来的交错路，顺便说一下，如果一条路 p 只包含一条边，那末不管这条边是属于匹配 M 的或是不属于匹配 M 的，p 一定是一条交错路（想想看，对不对？）．

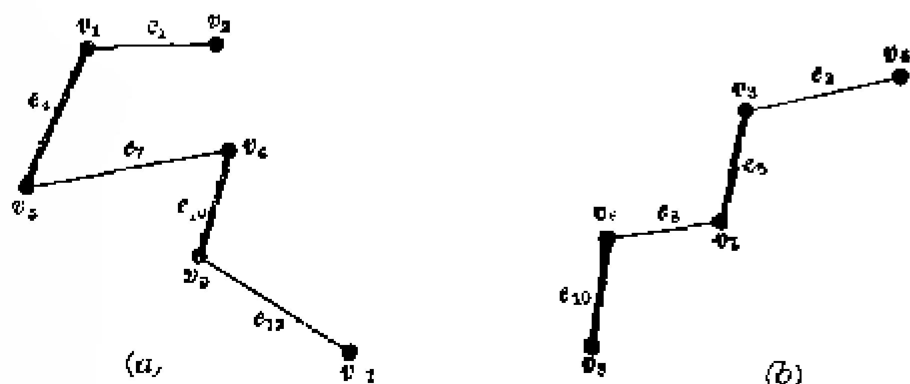


图 10.2

定义 10.3 两个端点都是未盖点的交错路叫做可扩充路．

拿图 10.2 中的两条交错路来说，(a) 中的那一条路的两个端点是 v_2 与 v_{11}，再回到图 10.1 中看一下，正好 v_2 与 v_{11} 都是未盖点，因此这条路是可扩充路，而 (b) 中的那条交错路不是可扩充路．另外，如果在图中发现了两个未盖点，它们之间有一条边相连，那末单单这一条边就组成了一条可扩充路了，为什么呢？因为前面已说过，单单这条边就组成一条交错

路，再加上它的端点都是未盖点，当然就是一条可扩充路了。

可扩充路有一个很重要的性质，就是：在可扩充路上，不属于匹配的边一定比属于匹配的边多一条。请大家自己检查一下这个性质对不对。

为什么采用"可扩充路"这样的一个名称呢？原来对于图 G 的一个匹配来说，如果能够找到一条可扩充路 p，那末这个匹配就一定可以改进成一个多包含一条边的匹配(也就是说，原来的匹配"扩充"了！)。改进的办法是：把 p 中原来属于匹配的边从匹配中去掉，而把原来不属于匹配的边加到匹配中去。按照我们以前的画图办法，我们常常把属于匹配的边画成粗边，那末上面的改进办法就可以简单地说成是：把路 p 上的粗边都改为细边，而把细边都改为粗边。前面说过，可扩充路 p 上，细边一定比粗边多一条。因此，变化后的匹配就恰好比原来的匹配多一条边。拿图 10.1 中的图 G 以及由粗线组成的匹配来说，图 10.2(a)中已经找出了一条可扩充路，因此，就可以按照刚才讲的"粗、细边对换"的办法把原来的匹配改进成一个多包含一条边的匹配了(见图 10.3，虚线表示可扩充路)。

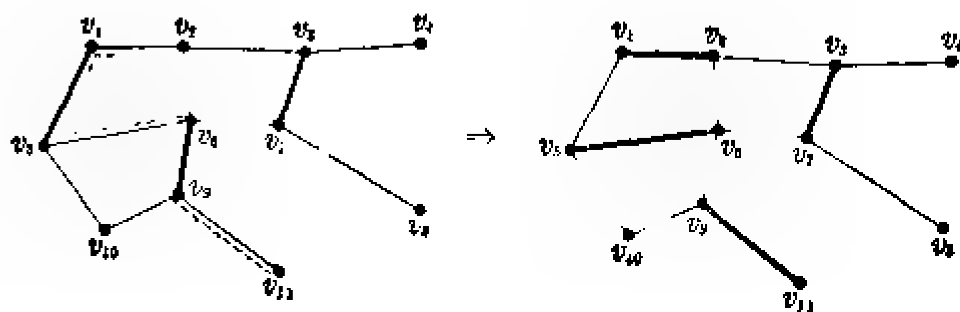


图 10.3

从上面讲的可以看出，我们已经有了一种把一个匹配改进成一个更加好一些的匹配的办法了。有了这种改进办法，

再结合着第八章讲的求网络的最大流的方法想一想，就不难得出一种求最大匹配的方法来了。这种方法的大概轮廓是：先任意找一个匹配 M_1，然后找找看，对于 M_1 来说，有没有可扩充路，如果有，就用刚才讲的粗、细边对换的办法把 M_1 改进成一个多包含一条边的匹配 M_2。然后再找找看，对于 M_2 来说，有没有可扩充路，有就再改进，……，一直到出现一个匹配 M_s，对于 M_s 来说，找不到可扩充路了，计算结束。

回想一下，在第八章中，我们曾经讲过，一般说来，用某一种改进的方法不能再改进的流，不一定是最大流（说不定还可以用别的方法再改进呢！）。对于匹配来说，也一样。用上面讲的计算方法得到的匹配 M_s，它只不过具有"用在可扩充路上粗、细边交换的办法不能再改进了"这样的特点，而这个特点不一定能保证它是最大匹配。不过，幸亏我们有下面的定理 10.1，根据这个定理，就可以肯定 M_s 是最大匹配了。

定理 10.1　设 M 是图 G 的一个匹配，并且对于 M 来说，不存在可扩充路，那末 M 一定是最大匹配。

这个定理的证明，这里就不写了，有兴趣的读者可以参看一些专门些的图论著作。

有了定理 10.1 以后，我们前面讲的求最大匹配的方法就比较完整了。我们再把这个方法的计算步骤总结一下：

步骤 1　用任何方法求出第一个匹配 M_1；

步骤 2　用定理 10.1 来检查现有的匹配是不是最大匹配，即看看是不是有可扩充路，如果没有可扩充路，那末计算结束（因为最大匹配已经找到了）；如果找到了可扩充路，转向步骤 3；

步骤 3　把现有的匹配用"粗、细边对换"的方法改进成

一个更好的(就是包含边更多的)匹配,转回步骤2.

这种方法的总的轮廓与第八章求最大流的方法是一样的. 因此,我们不必再多加解释了. 而主要来分析一下这三个步骤做起来是不是容易.

步骤1是很容易做的. 例如前面讲过,空集ϕ就是一个匹配,因此,我们可以就拿空集ϕ作为第一个匹配M_1. 另外,很明显,对于图G的任意一条边e_i来说,单单由一条边e_i组成的边集合也是一个匹配,因此,也可以任意取一条边e_i,而以$M_1=\{e_i\}$作为第一个匹配. 当然,如果可以用很简单的办法一眼就看出另一个边数更多的匹配M_1,也可以取它作为第一个匹配.

在找到了一条可扩充路以后,在这条路上用粗、细边对换的办法来改进一个匹配也是很容易的. 也就是说,步骤3也是容易做的.

但是步骤2却不容易做了. 这个情况倒也是和第八章的求最大流的算法相似的. 因此,我们将在下面两节中专门讨论一下步骤2应该怎样做,也就是说,怎样来找可扩充路.

习 题

给图10.1中的图G找一个最大匹配(计算时,以图上的粗边作为第一个匹配M_1,而用直接观察的办法找可扩充路).

2. 交 错 树

在定理10.1得到了证明以后,就有人提出过一种找可扩充路的方法,就是,对于图G的一个给定的匹配M来说,首

先看看有几个未盖点．如果没有未盖点或者只有一个未盖点，就不用找了，肯定不会有可扩充路了，如果未盖点超过一个，就任意取定一个未盖点 v_i，然后从 v_i 出发，用"搜索"的办法来找以 v_i 为一个端点的可扩充路．具体地说，就是从 v_i 出发，沿着一条交错路前进，如果在前进中遇到了另一个未盖点，那末就找到可扩充路了．如果遇不到未盖点，而又无法沿着交错路再前进了，就倒退回来，再沿着别的没有走过的路前进，…．拿图 10.4 中的图来说，有 3 个未盖点．例如我们先取定 v_3，从 v_3 出发先沿着 e_1 走到 v_2，到了 v_2 以后，不能沿着 e_3 走(因为这样就不是沿着一条交错路前进了)，而只能沿着 e_2 走到 v_1，到了 v_1 就没法再前进了，只好退回一步，退到 v_2，仍旧没有新的路可以走，又退到 v_3，从 v_3 沿着新的路 e_7 走到 v_8，接着又走到 v_9，又完了，再退回到 v_3，因为从 v_3 出发的两条边都走过了，这就说明以 v_3 为一端的可扩充路是不存在的．于是再另取一个未盖点，例如取 v_4，从 v_4 出发再进行搜索，…．

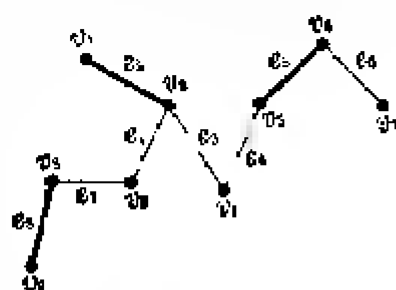


图 10.4

很显然，如果遇到一个比较复杂的图，这种方法是很难用的．这种方法的主要缺点，在于搜索时太不系统，有些边和顶点往往反复地走到过好几次(例如从 v_4 出发又可能走到 v_2 和 v_1)．

为了克服上面讲的搜索时不够系统这一缺点，并且设计一种系统搜索可扩充路的方法，我们还需要介绍几个概念．

首先回想一下，在第二章中，我们曾经讲过"由顶点 v_i 可达 v_j"这一概念，在寻找以未盖点 v_i 为一端的可扩充路时，我

们将用到一个类似的概念，就是：

定义 10.4 设 M 是图 G 的一个匹配，v_i 是取定的一个未盖点，如果存在着连接 v_i 与另一顶点 v_j 的交错路，就称由 v_i 交错可达 v_j（我们也认为 v_i 交错可达 v_i 本身）.

仍以图 10.4 为例．如果我们取定了未盖点 v_3，那末因为存在着交错路：

$$p=\{v_3, e_1, v_2, e_2, v_1\},$$

因此由 v_3 交错可达 v_1，当然，由 v_3 也交错可达 v_2，因为 p 中从 v_3 到 v_2 这一段也是交错路，不难看出，由 v_3 还交错可达 v_8 与 v_9.

有了定义 10.4，我们可以看出，如果发现了一个未盖点 v_i，而由 v_i 又交错可达 v_j，我们就找到了一条连接 v_i 与 v_j 的交错路，也就是一条可扩充路了．因此，要想找以 v_i 为一端的可扩充路，可以采取下面的办法．设法把由 v_i 交错可达的顶点一个一个地找出来，每找到一个，就检查一下它是不是未盖点，是的话可扩充路就找到了，如果已经把所有由 v_i 交错可达的顶点都找了出来，而其中没有一个是未盖点，就可以肯定以 v_i 为一端的可扩充路一定不存在了.

为了把由 v_i 交错可达的顶点都找出来，还要用到一个很重要的概念，叫做"以顶点 v_i 为根的交错树"．回想一下，在第五章中我们曾经讲过，如果 $G=[V, E]$ 是一个树，那末对于 G 的任意两个不同的顶点 v_i 与 v_j，在 G 中一定存在唯一的一条连接 v_i 与 v_j 的初级路（第五章第 2 小节习题 2），下面讲交错树的定义时要用到这个结论.

定义 10.5 设 M 是图 $G=[V, E]$ 的一个取定的匹配，T 是图 G 的一个子图，如果 T 具有下述性质：

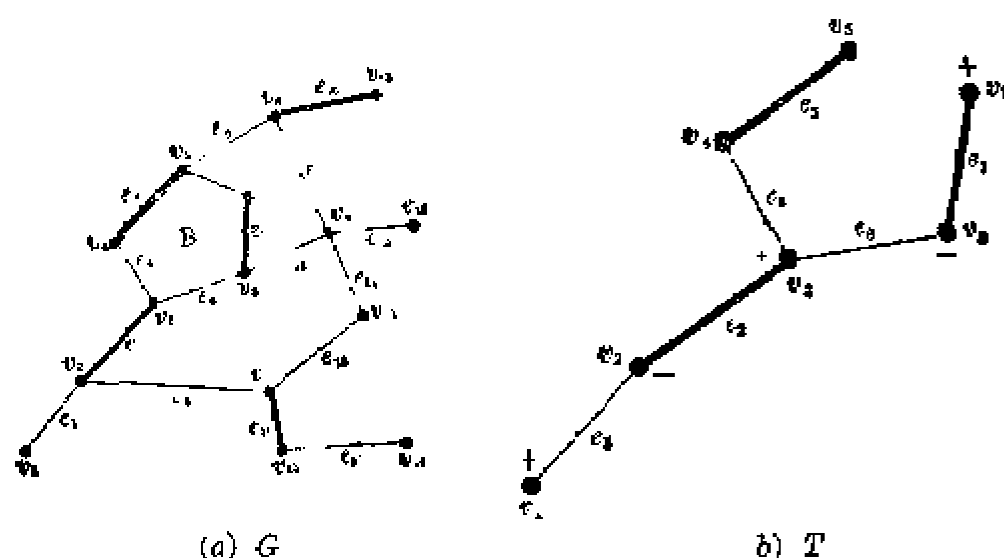
(1) T 是一个树;

(2) T 中存在一个顶点 v_i, 它是未盖点;

(3) 对于 T 的任意一个不同于 v_i 的顶点 v_j 来说, T 中连接 v_i 与 v_j 的唯一初级路是交错路.

那末称 T 是一个以 v_i 为根的交错树.

以图 10.5(a)中的图 G 为例(粗边代表一个匹配), 图 10.5(b)中画的子图 T 就是一个以 v_1 为根的交错树.



(a) G　　　　(b) T

图 10.5

由定义 10.5 中的(3), 不难看出: 交错树上的每一个顶点 v_j 都是由 v_i 交错可达的顶点. 另外, 要具体地把连结 v_j 与根 v_i 的交错路找出来也不难, 因为在 T 中连接 v_i 与 v_j 的路是唯一的, 而由定义 10.5 的(3), 这条路就是要找的交错路.

因此, 想要把由 v_i 交错可达的顶点都找出来, 可以采用不断扩大一个交错树 T 的办法. 开始时, 令交错树 T 只含一个顶点, 就是取定的未盖点 v_i, v_i 是 T 的根. 以后逐步扩大 T, 每扩大一次, T 中新增加的顶点就是新找到的由 v_i 交错可达的顶点.

为了讲怎样扩大一个交错树，还要介绍一个概念．请大家看一下图 10.5(b) 中的以 v_1 为根的交错树 T，它的每一个顶点旁边或者标了一个“+”号，或者标了一个“−”号．这样做主要是为了说明 T 的顶点可以分成两类，一类叫做外点，就是旁边标“+”的顶点，一类叫做内点，就是旁边标“−”的顶点．那末我们是按照什么原则来区别这两种点的呢？是这样的：首先，我们规定根 v_i 是外点；其次，对于 T 其他顶点 v_j 来说，为了区分它是内点还是外点，应该先把连结根 v_i 与 v_j 的交错路 p 找出来，然后数数 p 包含几条边，如果 p 包含奇数条边，v_j 就是一个内点，而如果 p 包含偶数条边，v_j 就是一个外点．仍旧拿图 10.5(b) 中的 T 来说，连接根 v_1 与 v_4 的路包含 3 条边，所以 v_4 是内点，而连接 v_1 与 v_5 的路包含 4 条边，所以 v_5 是外点，请大家检查一下，其他顶点旁边标的“+”与“−”标得对不对．

关于外点和内点，有一个重要的性质，它在后面经常要用到，就是：

引理 10.1 设 T 是一个以 v_i 为根的交错树，v_j 是 T 的一个不同于 v_i 的顶点，p 是 T 上连接 v_i 与 v_j 的交错路，那末：

(1) 如果 v_j 是外点，p 上与 v_j 关联的边一定属于匹配 M．

(2) 如果 v_j 是内点，p 上与 v_j 关联的边不属于匹配 M．

证明 如果 v_j 是外点，那末 p 应该包含偶数条边，现在从 v_i 出发沿着 p 向 v_j 走，第一条边一定不属于 M(因为 v_i 是未盖点)，所以第二条边应该属于 M，而第三条边又不属于 M，第四条边又属于 M，…，因为 p 包含偶数条边，所以最后

的那一条边，也就是与 v_j 关联的边，一定属于 M.

v_j 是内点的情况是完全相似的. 证完.

请大家结合着图 10.5(b) 检验一下，引理 10.1 对不对.

有了以上的准备，就可以讲扩大一个以 v_i 为根的交错树的办法了. 这个办法是以下面的引理为基础的.

引理 10.2 设 v_i 是 T 的一个外点，e 是不属于 T 的一条边，如果 e 与 v_j 关联，并且 e 的另一个端点 v_k 也不属于 T，那末由 v_i 交错可达 v_k.

证明 因为 v_j 是外点，由引理 10.1，T 上连接根 v_i 与 v_j 的交错路 p 上最后的一条边(就是与 v_j 关联的边)属于匹配，容易看出，在 p 上再加一条边 e，就得到连接 v_i 与 v_k 的交错路(见图 10.6)，因此由 v_i 交错可达 v_k. 证完.

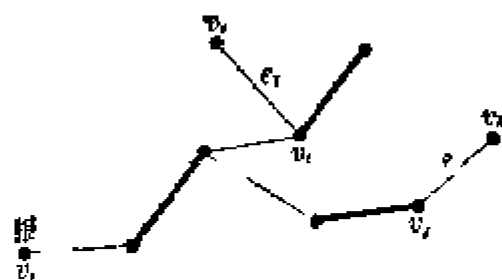


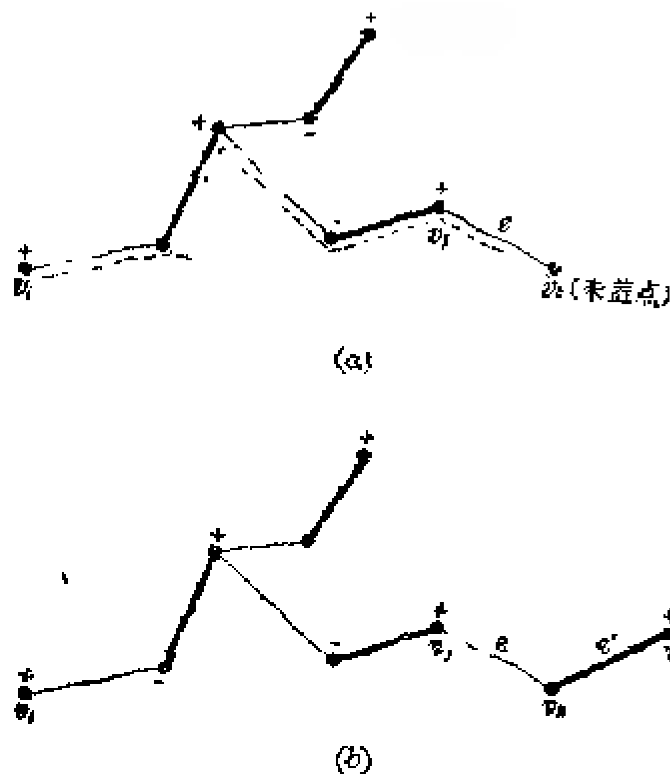
图 10.6

注意，引理 10.2 中，"v_j 是外点"这一条件是很重要的，如果发现一条不属于 T 的边 e_1 与 T 的一个内点 v_t 关联，并且 e_1 的另一个端点 v_s 也不属于 T，是不能保证由根 v_i 交错可达 v_s 的(见图 10.6).

有了引理 10.2，就可以得到一种扩大交错树的办法了，就是，看看图 G 中有没有与交错树 T 的外点关联而不属于 T 的边 e，如果有，就再看看 e 的另一个端点 v_k 是不是属于 T，如果 v_k 不属于 T，就可以把 e 和 v_k 都加到 T 中去，使 T 扩大. 具体扩大的时候，还可以分两种情况:

(1) v_k 是未盖点，这时，把 e 与 v_k 加到 T 中去后，T 中连接根 v_i 与 v_k 的交错路是一条可扩充路[见图 10.7(a)].

(2) v_k 不是未盖点，也就是说，有一条属于匹配 M 的边 e' 与 v_k 关联，这时，在把 e 与 v_k 加到 T 中去后，接着还可以把 e' 以及它的端点加到 T 中去，因为很显然从 v_i 也交错可达 e' 的另一个端点 v_l[见图 10.7(b)]。另外，还容易看出，v_k 应该是内点，而 v_l 是外点。



(a)

(b)

图 10.7

因此，在发现满足引理 10.2 的边 e 时，或者可以找到一条可扩充路，或者可以把 T 扩大，使 T 增加两条边和两个顶点。

让我们结合着图 10.5(a) 中的图 G[见图 10.8(a)] 来看看怎样用这种办法扩大交错树。取未盖点 v_1，一开始，T 只包含顶点 v_1，v_1 是外点[见图 10.8(b)]，容易看出，边 e_1 符合引理 9.2 的条件，并且 e_1 的另一个端点不是未盖点，因此，可以把边 e_1，e_2 及顶点 v_2、v_3 加到 T 中去，扩大后的交错树仍

记作 T[见图 10.8(c)]，接着，又可以把边 e_4、e_5 以及顶点 v_4、v_5 加到 T 中去；还可以把边 e_6、e_7 与顶点 v_8、v_7 加到 T 中去[见图 10.8(d) 与(e)]. 最后，还可以把边 e_9、e_{10} 以及顶点 v_6、v_{15} 加到 T 中去，把 T 扩大成图 10.8(f).

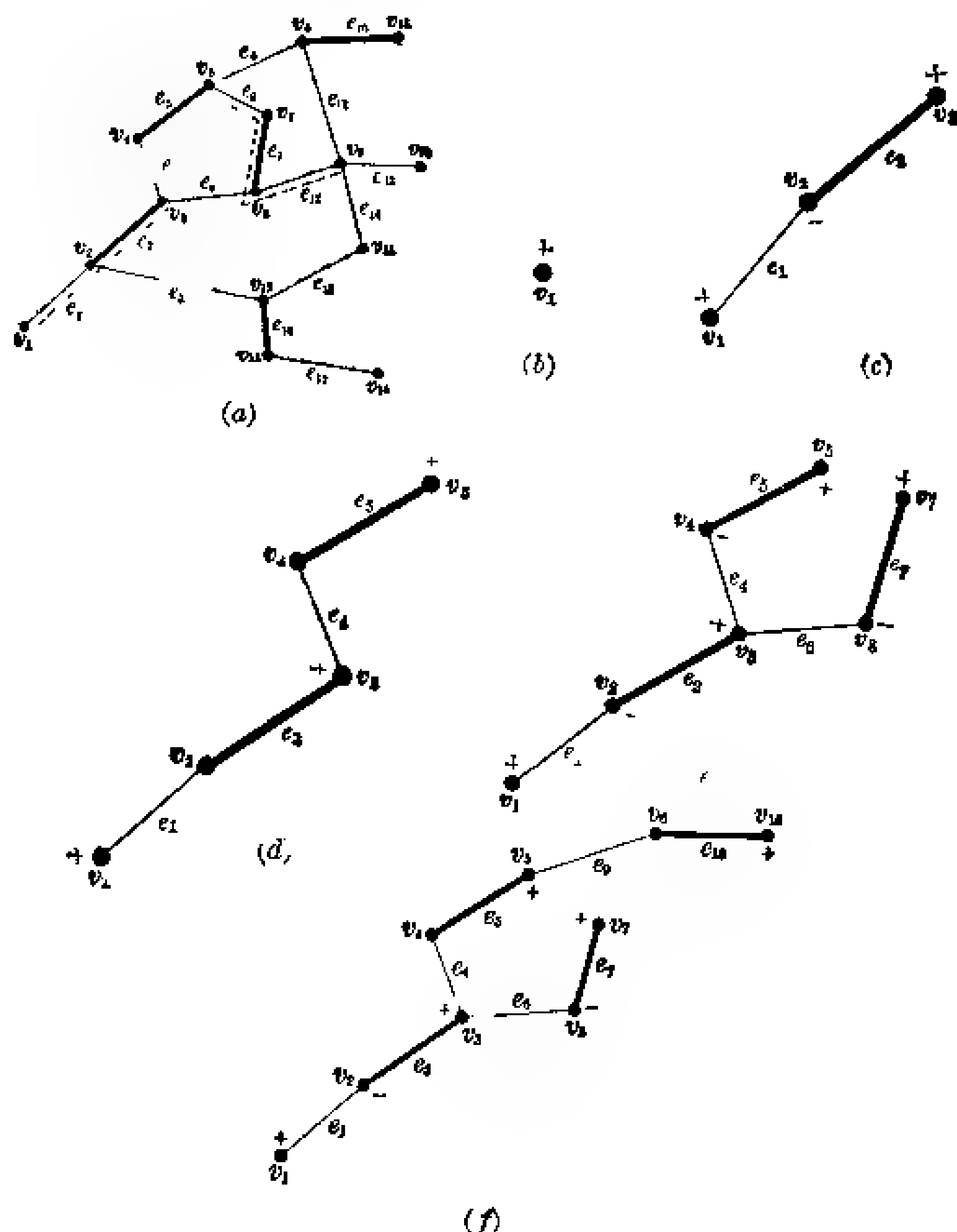


图 10.8

对于图 10.8(f)中的交错树，不能再用引理 10.2 的办法

扩大了，因为找不到满足引理10.2的条件的边e。但是能不能就此断定以v_1为一端的可扩充路一定不存在了呢？还不能。事实上，以v_1为一端的可扩充路是存在的，图10.8(a)中用虚线标出的那条路就是一条连接v_1与v_8的可扩充路。为了把这样的可扩充路找出来，还需要一个重要的概念，让我们在下一节再讲吧。

习　题

给图10.5(a)中的图G找一个以v_0为根的交错树，并在顶点旁边标上"+"或"−"号来说明哪些顶点是外点，哪些顶点是内点。

3. 带花的树

让我们再看一下图10.8(f)中的交错树T（见图10.9）。上一节最后讲过，用引理10.2讲的办法不能再使T扩大了。但是我们却可以用另一种办法来继续扩大T。

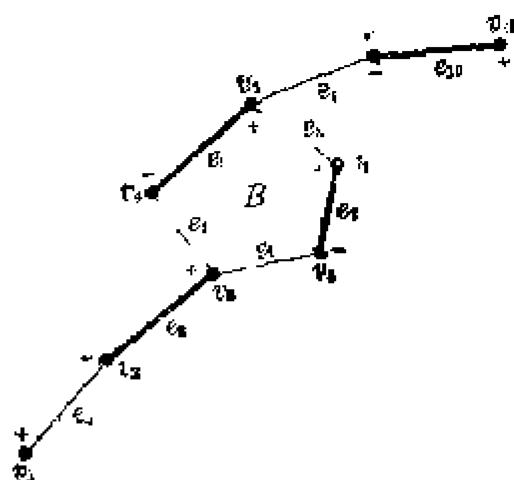


图　10.9

请大家再看一下图10.8(a)中的图G，我们看到，图G中的边e_8，它不属于T，而它的端点是T的两个外点v_5与v_4，在出现这种情况时，我们把边e_8也加到T中去，使T又得到扩大，即又使T增加了一条边（但是顶点没有增加）。

应该注意，把e_8加到T中去后，T就不再是树了，因为

它包含一个圈，就是图 10.9 中的 B，这样的子图今后叫做带花的树。

上面我们只是通过一个例子来讲了什么叫带花的树，一般情况也完全一样，就是如果发现了一条不属于交错树 T 的边 e，e 的两个端点都是 T 的外点，那末把 e 加到 T 中去得到的图（仍记作 T）叫做带花的树。

带花的树的特点是，它恰好有一个圈．今后我们就把这唯一的圈叫做“花”．请大家记住，“花”就是一个圈，而且不难看出，这种圈一定包含奇数条边（图 10.9 中的花 B 包含 5 条边，一般情况请大家自己证明一下）．

回想一下，我们以前讲过，T 中的顶点都是由 v_1 交错可达的，而扩大 T 的目的主要是扩大由 v_1 交错可达的点的范围．现在把 e_8 加到 T 中去，只不过使 T 增加了一条边，并没有增加顶点，又有什么作用呢？

我们就结合着图 10.9 来解释一下把 e_8 加到 T 中去有什么好处．在图 10.9 中，v_8 是花上的一个内点，从图 10.8(a) 可以看出，有一条不属于 T 的边 e_{12} 与 v_8 关联，e_{12} 的另一个端点 v_9 也不属于 T．如果 v_8 是 T 的外点，那末边 e_{12} 就满足引理 10.2 的要求，但是现在 v_8 是内点，前面讲过，这时不能保证从根 v_1 交错可达 v_9．不过在把连结外点 v_5 与 v_7 的边 e_8 加到 T 中去以后，情况就不同了，从图 10.10 可以看出，现在我们可以找到

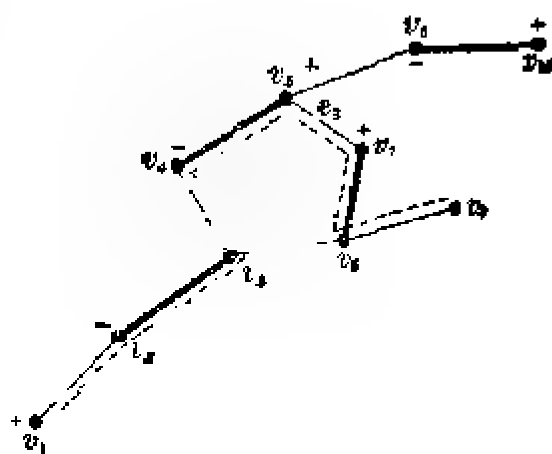


图 10.10

连接 v_1 与 v_9 的交错路了，就是图 10.10 中用虚线标出的那一条. 不难想象，如果有一条不属于 T 的边 e 与花上的内点 v_4 关联，并且 e 的另一个端点 v_k 也不属于 T，那时我们也可以找出一条连结 v_1 与 v_k 的交错路（想想看，这条交错路应该经过哪些顶点到达 v_k）. 从上面的讨论可以看出，花上的内点都具有外点的作用，也就是说，只要发现花中任意一个顶点（不管是外点还是内点）与一条不属于 T 的边 e 关联，而且 e 的另一个顶点 v_k 也不属于 T，就可以肯定由 v_1 交错可达 v_k，从而可以象上一节讲的那样把 e 与 v_k 加到 T 中去，使 T 扩大.

讲到这里，大家可能会感到问题变得复杂起来了：一般说来，与内点关联的边不必考虑，而对于花来说，与内点、外点关联的边都要考虑. 有没有办法使这个复杂情况变得简单一些？

有人想出了一个办法（这也是整个这个计算方法中最精彩的地方），就是，在出现带花的树时，把花"收缩"成一个顶点. 这样一收缩大有好处. 首先，带花的树收缩后又变成一棵交错树了，例如图 10.11 中画的就是把图 10.10 中的带花的

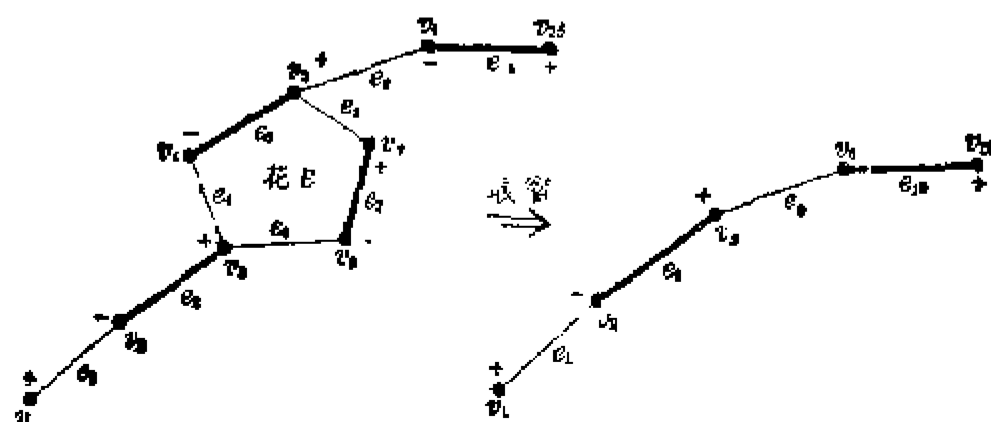


图 10.11

树收缩成一个交错树的过程. 图 10.11 左边的花 B, 在收缩后变成了一个顶点, 就是右边的顶点 v_B, 因为花 B 上的每一个顶点都具有外点的作用, 所以在收缩后的交错树上, v_B 是一个外点.

"收缩"的办法我们已经不是第一次遇到了. 在第七章讲求最小树形图时也遇到过. 和那时一样, 所谓把花 B 收缩成一个顶点应理解为: 我们是把原来图 10.8(a) 中的图 G 中的圈 B 收缩成一个顶点 v_B 了, 这样一收缩, 图 G 就变成另一个图 G_1 了. 在从 G 收缩成 G_1 的过程中, G 中所有两个端点都属于 B 的边都被收缩掉了, 而 G 中一个端点属于 B, 另一个端点不属于 B 的边都成为 G_1 中以 v_B 为一端的边了, 例如图 10.12 中画的就是从 G 收缩成 G_1 的过程. G 中的边 e_2, e_6, e_{12} 都是一端属于 B 而另一端不属于 B 的边, 在 G_1 中它们都成为以 v_B 为一端的边了.

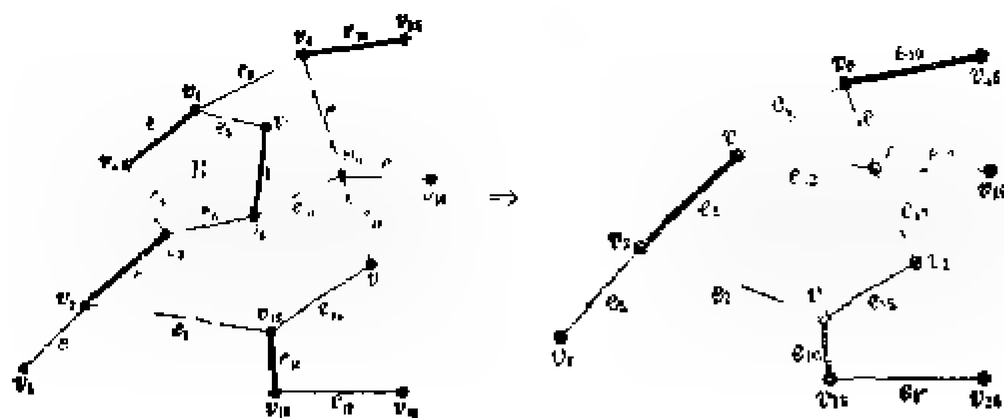


图 10.12

在把 G 收缩成 G_1 的过程中, G 的子图也就变成 G_1 的子图了, 特别是, G 中的带花的树 T 恰好收缩成 G_1 的一个交错树 T_1.

把上面讲的总结一下, 就是, 在发现了一条不属于交错树

T 并且连结交错树上两个外点的边 e 时，我们就做下面两件事.

(i) 把 e 加到交错树 T 中去，使 T 成为一个带花的树.

(ii) 把原图 G 中的花 B 收缩成一个顶点 v_B，使 G 变成另一个图 G_1，而 G 的带花的树 T 就收缩成 G_1 的一个交错树 T_1 了.

接下去，我们就在 G_1 中找以根 v_1 为一端的可扩充路，当然，我们不必从头做起，而就在 T_1 的基础上来找，也就是设法在 G_1 中扩大 T_1，在扩大的过程中，如果找到了可扩充路，这时，我们也就找到了 G 的可扩充路了．为什么呢？让我们仍旧结合前面的例子来讲，例如图 10.11 右面的 T_1 就是图 10.12 右面 G_1 的一个交错树，从图 10.12 可以看出，G_1 中有一条连接 v_B 与 v_9 的边 e_{12}，v_B 是 T 的外点，v_9 是未盖点．因此，在把 e_{12} 与 v_9 加到 T_1 中去后，我们在 G_1 中就找到了一个可扩充路，即图 10.13 左边的那条路，这条路经过收缩点 v_B，这时，应该把 v_B "展开"成花 B(见图 10.13，展开后很容易找到原图 G 的一条以 v_1 为一端的可扩充路了，也就是图中虚线表出的那一条(这也就是上一节最后指出的图 10.8(a) 中的那条可扩充路)．在 G_1 中扩充交错树 T_1 时，如果出现

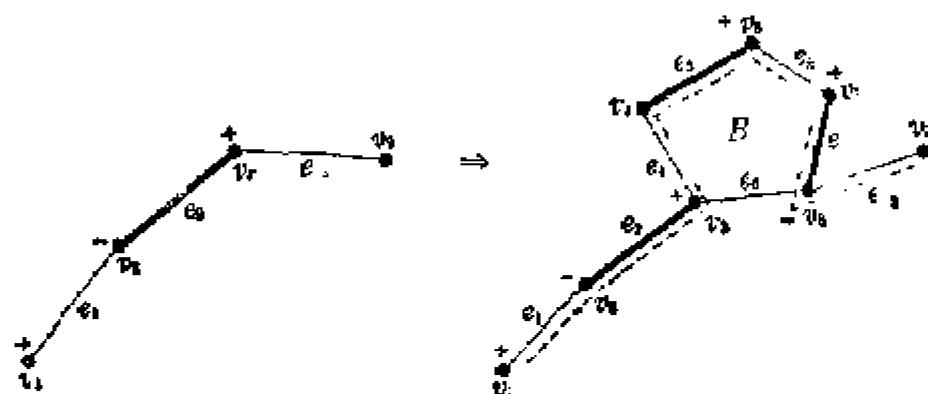


图 10.13

了引理 10.2 中的情况，就扩大 T_1. 而如果又发现连接两个外点的边时，就再收缩，即把 G_1 收缩成另一个图 G_2，这时 T_1 就收缩成 G_2 的交错树 T_2，如果在 G_2 中找到了可扩充路，就要两次把收缩点展开成花，来找 G 中的可扩充路.

讲到这里，找以未盖点 v_i 为一端的可扩充路的方法基本上讲完了，总结起来，就是：用上一节引理 10.2 中讲的办法与这一节讲的办法不断的扩大交错树 T，当然在遇到带花的树时就应该收缩，在扩大 T 的过程中，如果发现可扩充路，那末我们的目的就达到了. 如果一直没有发现可扩充路，而又出现了无法用上一节和这一节讲的办法再扩大 T 的情况，则可以根据下面的定理断定一定不存在以 v_i 为一端的可扩充路.

定理 10.2 设 v_i 是一个未盖点，并且在扩大以 v_i 为根的交错树时(可能是在经过几次收缩以后)，出现了既不能用引理 10.2 的办法又不能用这一节讲的办法再扩大的情况，那末，一定不存在以 v_i 为一端的可扩充路.

这个定理的证明就从略了.

分析一下定理 10.2 的结论，就可以看出上一节与这一节讲的找可扩充路的方法的优点了. 因为扩大以未盖点 v_i 为根的交错树的结果，或者是找到了可扩充路，或者是出现了定理 10.2 的情况，从而可以断定以 v_i 为一端的可扩充路一定不存在. 决不会出现模棱两可的情况.

在出现了定理 10.2 的情况后，我们应该再取一个不同于 v_i 的未盖点 v_j，然后来扩大以 v_j 为根的交错树. 当然这样做的结果或者是找到了可扩充路，或者是断定以 v_j 为一端的可扩充路不存在. 然后再取一个没有考虑过的未盖点，……. 如果所有未盖点都考虑了一遍，每次都是出现定理 10.2 的情

边，那末就可以肯定不会有可扩充路，从而根据定理 10.1，可以断定现有的匹配是最大匹配了。

习 题

证明花一定包含奇数条边。

4. 开花算法总结、例子

回想一下第1小节最后讲的求最大匹配的三个步骤，可以看出，到此为止，求最大匹配的方法已经讲完了，因为这种计算方法中用到收缩花和展开花，因此有人给这种算法取了一个名字，叫"开花算法"（英文名称为 Blossom Algorithm）。

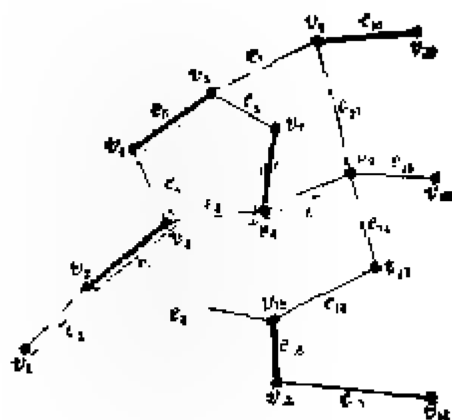


图 10.14 G

作为这个算法的总结，我们来计算一个例子。我们仍旧考虑前面讲过的图 10.8(a) 中的图 G，并且就取这个图中的粗边作为第一个匹配 M_1（见图 10.14）。

对于匹配 M_1 来说，有5个未盖点，就是 v_1，v_9，v_{10}，v_{11} 和 v_{14}，在第2小节与第3小节中，我们已经用扩大以未盖点 v_1 为根的交错树的办法找到了一条可扩充路，即图 10.14 中用虚线标出的那一条，在这条可扩充路上可以用粗、细边对换的办法把匹配 M_1 改进成图 10.15 中用粗边表示的匹配 M_2。

对于匹配 M_2 来说，还有3个未盖点 v_{10}，v_{11} 和 v_{14}。现在就取定 v_{10}，然后来扩大以 v_{10} 为根的交错树 T。开始时，T 只

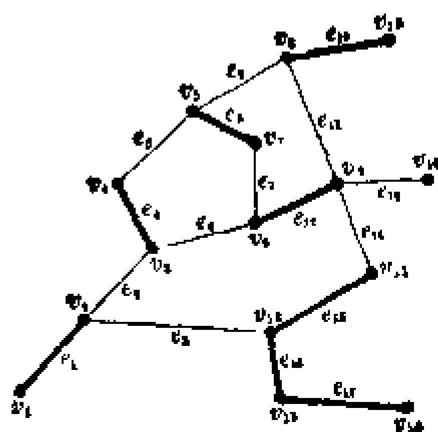
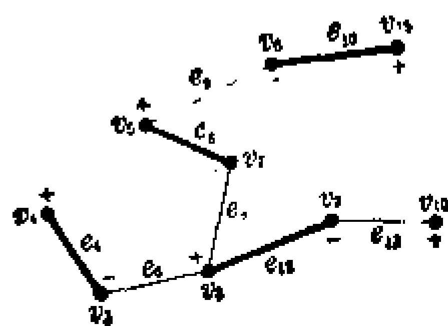
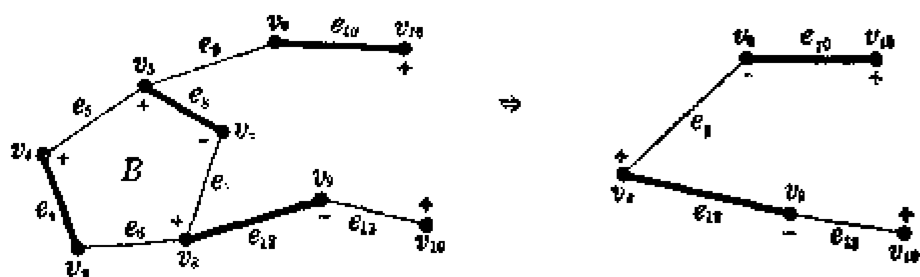

(粗边为匹配 M_2)

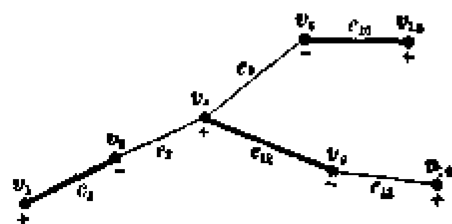
图 10.15


图 10.16 T

含 v_{10} 一个顶点，连续用引理 10.2 的方法将 T' 扩大 4 次，可以得到图 10.16 中的交错树 T'. 现在 e_5 是不属于 T' 而连结 T 的两个外点的边，按照第 3 小节所讲的，应该先把 e_5 加到 T 中去使 T 成为带花的树，然后把花 B 收缩成一个顶点(见图 10.17)，从而把带花的树 T 又变成一个交错树 T_1. 当然，这


图 10.17

时我们也应把图 G 中的圈 B 收缩成一个顶点 v_B，使 G 变成另一个图 G_1，G_1 这里就不画了．不难看出，收缩后边 e_2 与 T_1 的外点 v_B 关联，并且 e_2 的另一个端点 v_2 不属于 T_1，因此 T_1 又可以扩大，即再增加两条边与两个顶点(见图 10.18). 现在，不管用第 2 小节引理 10.2 的方法还是第 3 小节的方法都无法再扩大 T_1 了(请大家仔细


图 10.18

检查一下，是不是这样?）．因此，按照定理 10.2，可以断定以 v_{10} 为一端的可扩充路一定不存在．因此我们应该再取一个未盖点，例如取 v_{11}，然后来扩大以 v_{11} 为根的交错树 T．不难看出，可以把边 e_{15}，e_{16} 以及顶点 v_{12}、v_{13} 加到 T 中去，接下来，又可以把边 e_{17} 与顶点 v_{14} 加到 T 中去，但是因为 v_{14} 是未盖点，所以我们已经找到可扩充路了（见图 10.19）．在这条可扩充路上用粗、细边对换的办法可以把图 10.15 中的匹配 M_2 改进成图 10.20 中的匹配 M_3．

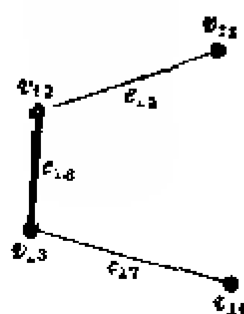
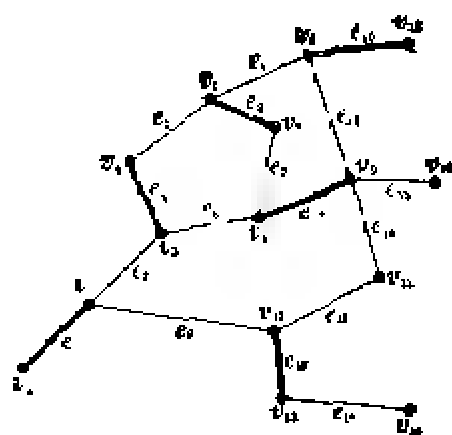


图 10.19



（粗边为匹配 M_3）

图 10.20

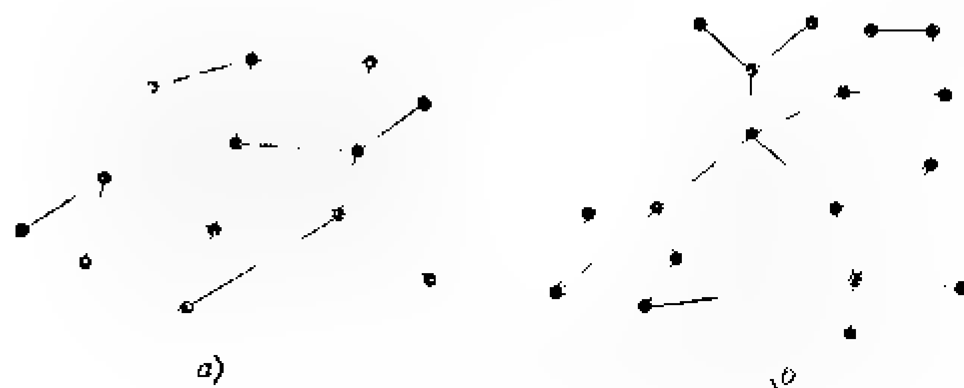
对于匹配 M_3 来说，只有一个未盖点 v_{10}，因此，M_3 一定是一个最大匹配了．

可以证明，这一章讲的计算方法是有效的，不过，这里就不仔细讲了，留给大家作为习题吧．

习　　题

1　请考虑一下，如果把第 1～4 六节中讲的算法用到二分图上去，得到的算法与上一章中讲的以求网络的最大流为基础的算法有什么关系?

2. 给下述两个图各找一个最大匹配.



3. 试试看，证明这一章讲的求最大匹配的计算方法是有效的.

十一、最小边复盖问题

1. 什么是最小边复盖问题

这一章我们将再介绍一个图论中的极值问题，它与上一章讲的最大匹配问题有密切的关系。先来介绍一下，什么叫最小边复盖问题。

定义 11.1 设 $G=[V, E]$ 是一个无向图，$C\subseteq E$ 是 G 的若干条边的集合，如果 G 的每一个顶点都是 C 中某一条边的端点，就称 C 是图 G 的一个边复盖。

例如图 11.1(*a*)、(*b*)、(*c*)的图中，粗边组成的集合分别是这三个图的边复盖。

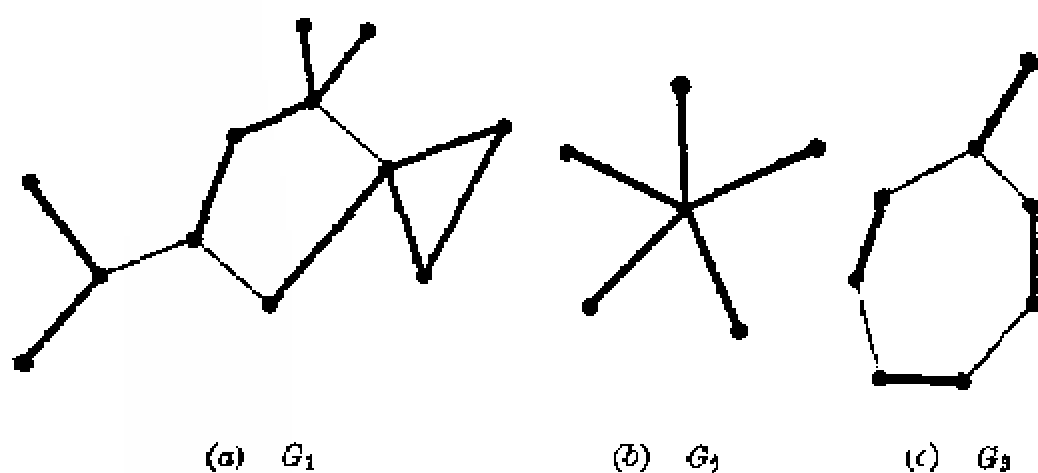


图 11.1

上一章介绍未盖点的概念时曾经说过，我们可以认为一

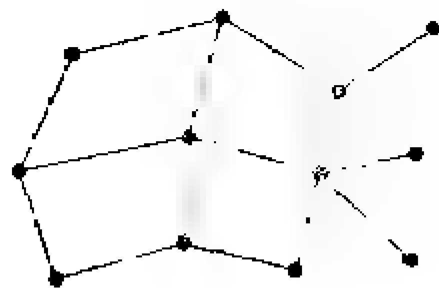
条边"盖住了"它的两个端点，在这种意义下，边复盖就恰好是把图 G 的所有顶点都盖住的边集合.

因为一条边只能盖住两个顶点．因此，如果一个图有 10 个顶点，那末它的每一个边复盖至少包含 5 条边，如果有 11 个顶点，那末边复盖就至少包含 6 条边了．一般说来，如果图 G 有 n 个顶点，那末边复盖包含的边不会少于 $\frac{n}{2}$ 条.

是不是任何图都有边复盖呢？不一定！因为只要图中有孤立顶点，就是不和任何边关联的顶点，这个图就不可能有边复盖了．不过很容易看出，只要图 G 没有孤立顶点，G 就一定有边复盖．因为这时，只要令 $C=E$，也就是说令 C 为图中所有边的集合，C 就是一个边复盖.

定义 11.2 设 $G=[V, E]$ 是一个没有孤立顶点的图，我们把 G 的包含边数最少的边复盖叫做最小边复盖.

所谓最小边复盖问题，指的就是：对于给定的图 G，要求把 G 的最小边复盖找出来.


图 11.2 G

最小边复盖问题是有一定的实际意义的．例如我们把图 11.2 中的图 G 中的顶点看成是一些村庄，每条边看成是一段公路．如果在一段公路上放上一辆消防车，就可以把这段公路两端的两个村庄控制起来了，现在问至少用几辆消防车才能把所有村庄全控制住？这些消防车应该放在哪些公路上？不难看出，上述问题可以归结为求图 G 的最小边复盖问题.

习　题

用直接观察的办法分别给图 11.1 和 11.2 中的每个图各找一个最小边复盖.

2. 求最小边复盖的方法

现在就来讲求一个图的最小边复盖的方法. 下面大家将要看到, 由于最小边复盖问题与上一章讲的最大匹配问题之间有着很密切的联系, 从而, 当要求图 G 的最小边复盖时, 只要先求出 G 的一个最大匹配 M, 然后在 M 的基础上稍加修改, 就可以得出 G 的最小边复盖了.

下面的两个引理说明了最小边复盖问题与最大匹配问题之间的关系. 在以下的讨论中, 都假设图 G 没有孤立顶点, 即 G 有边复盖.

引理 11.1 设图 G 有 n 个顶点, 又设 G 的最大匹配 M 由 s 条边组成, 那末 G 一定有一个由 $n-s$ 条边组成的边复盖.

证明 因为匹配 M 包含 s 条边, 并且这些边都没有公共的端点, 因此这 s 条边盖住了 $2s$ 个顶点. 也就是说, 对于 M 来说, 有 $n-2s$ 个未盖点. 现在我们从 M 出发来构造一个图 G 的边复盖 C': 首先, 把 M 中的边都放在 C' 中, 然后, 对于每一个未盖点 v_i, 任意取一条与 v_i 关联的边 e_i, 把它放在 C' 中. 对于 $n-2s$ 个未盖点来说, 这样得到的 $n-2s$ 条边互相不会相同(因为如果对于不同的未盖点 v_i 与 v_j, 得到的边相同, 例如都是 e, 那末把 e 加到 M 中去, 得到的仍是一个匹配, 与 M 是最大匹配就矛盾了), 因此 C' 包含有 $s+(n-2s)=n-s$

条边．另外，容易看出 C' 确实是一个边复盖．这就证明了存在一个由 $n-s$ 条边组成的边复盖．证完．

看一下图 11.3，对引理 11.1 的证明就会更加清楚了．图 11.3(a) 中的粗线表示一个由 2 条边组成的最大匹配 M（请大家用上一章讲的方法检查一下，M 是不是最大匹配），它盖住了 4 个顶点，图 G 共有 7 个顶点，因此还有 3 个未盖点，就是 v_4、v_6、v_7．再取 3 条边 e_2、e_3、e_7，分别与 v_4、v_6、v_7 关联，和 M 合在一起，就得到了图 11.3(b) 中的那个包含 $n-s=5$ 条边的边复盖了．

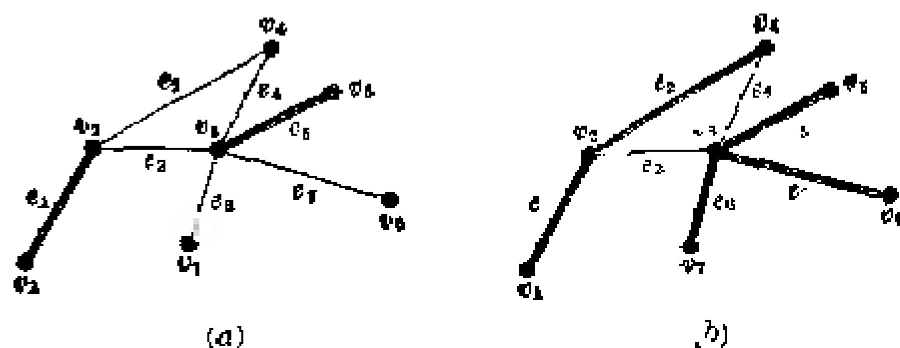


图 11.3

引理 11.2 设图 G 有 n 个顶点，并且有一个由 t 条边组成的最小边复盖 C，那末 G 一定有一个由 $n-t$ 条边组成的匹配 M'．

证明 我们从 C 中去掉一些边，来得到一个匹配，去边的办法如下：设 G 的顶点是 v_1、v_2、…、v_n．先看顶点 v_1，如果 C 中与 v_1 关联的边多于一条，例如有 p 条，就去掉其中的 $p-1$ 条，使 C 中只留下一条边与 v_1 关联．为简单起见，把去掉边后余下的边集合仍记作 C．注意，每去掉一条边，就会产生一个（相对于 C 来说的）未盖点．为什么呢？请看一下图 11.4，例如在考察 v_1 时，发现有 3 条边 e_1，e_2，e_3 与它关联，这

时，这三条边的另一个端点一定只能和C中的一条边关联，因为如果象图 11.4 画的那样，顶点v_4除了和边e_3关联外，还和边e_4关联，那时，把e_3从C中去掉，得到的将仍旧是一个边复盖（因为e_3的两个端点都能被C中其他的边盖住），而这就与C是最小边复盖矛盾了。考察完v_1以后，再考察v_2，如果与v_2关联的边多于一条，则再去掉几条，仍旧只留下一条与v_2关联的边，……，直到把所有边都考察过一遍为止。最后剩下的边集合就称为M'，很显然，M'是一个匹配。

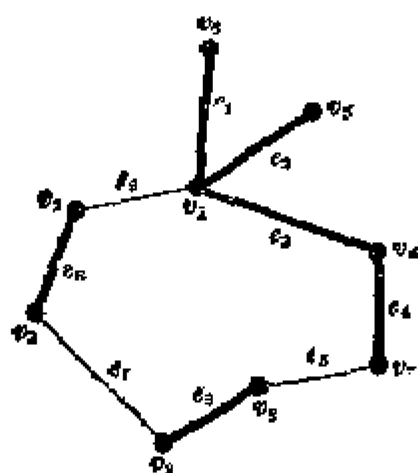


图 11 4

假设从C变到M'的过程中，一共去掉了y条边，因为C中原来有t条边，所以M'中应该剩下$(t-y)$条边。另外按照刚才讲的，每去掉一条边，就产生一个未盖点，因此，总共应该产生y个未盖点，我们的目的是证明存在一个由$n-t$条边组成的匹配，而我们现在已经找到了一个包含$t-y$条边的匹配M'，不过我们可以证明：

$$t-y=n-t.$$

为什么呢？因为G中共有n个顶点，现在我们找到了一个由$t-y$条边组成的匹配M'，它盖住了$2(t-y)$个顶点，另外我们还知道共有y个未盖点，因此，应该有：

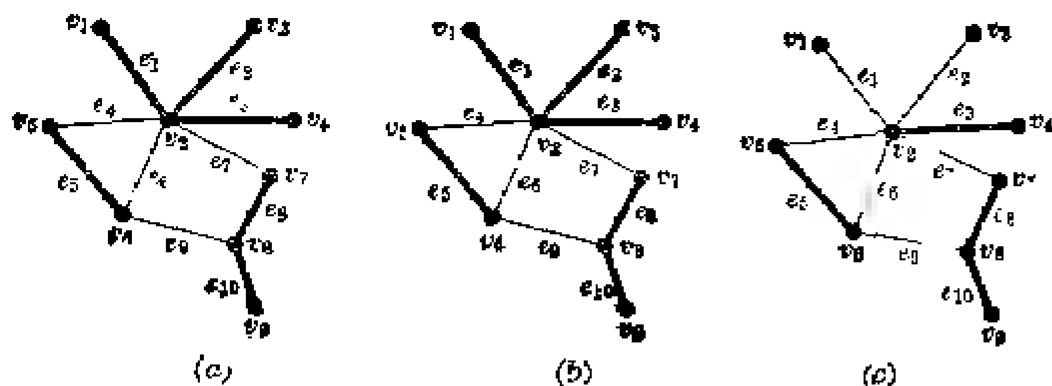
$$n=2(t-y)+y.$$

即
$$n=2t-y.$$

移项得
$$n-t=t-y.$$

这就证明了匹配M'由$n-t$条边组成。证完。

看一下下面的例子就会对引理 11.2 的证明过程更清楚了。例如对于图 11.5 中的图 G 来说，我们已经找到了一个由 6 条边组成的最小边复盖 C[见图 11.5(a)中的粗边]。考察 v_1，C 中只有一条边与 v_1 关联，不必去边。再考察 v_2，C 中有 3 条边与 v_2 关联，应该去掉其中 2 条，例如就去掉 e_1 与 e_2，C 中还剩下 4 条边[见图 11.5(b)]，从图中可以看出，去掉 2 条边就产生了 2 个未盖点[相对于图 11.5(b)中去掉边后的 C 来说]。再考察 v_3，没有 C 中的边与 v_3 关联，不用去边。对于 v_4，v_5，v_6，v_7，也不用去边。对于 v_8，有两条边与它关联，又应该去掉一条，例如去掉 e_{10}，现在 C 中只剩下 3 条边了[见图 11.5(c)]，再考察 v_9，不必去边。因此最后剩下一个由 3 条边组成的匹配。注意，图 G 中有 9 个顶点，最小边复盖含有 6 条边，因此引理 11.2 要求的正是一个由 9－6＝3 条边组成的匹配。



(a) (b) (c)

图 11.5

有了上面的两个引理，就可以证明下面的主要定理了。

定理 11.1 设图 G 有 n 个顶点，并且 G 的最大匹配 M 包含 s 条边，那末 G 的最小边复盖一定包含 $n-s$ 条边。

证明 首先，由引理 11.1，我们可以造出一个边复盖 C'，C' 包含 $n-s$ 条边。

现在证明 C' 就是最小边复盖.

用反证法，如果 C' 不是最小边复盖，也就是说，最小边复盖是另一个边复盖 C，并且 C 包含的边数 $t<n-s$. 由引理 11.2，存在一个包含 $n-t$ 条边的匹配 M'.

但是，因为
$$t<n-s,$$
所以
$$n-t>n-(n-s)=s,$$
也就是说 M' 包含的边($n-t$ 条)比 M 包含的边(s 条)还要多，这就与 M 是最大匹配矛盾了.

因此 C' 是最小边复盖，这就证明了最小边复盖包含 $n-s$ 条边. 证完.

从定理 11.1 可以看出，在找到了图 G 的最大匹配 M 后，用引理 11.1 的证明过程中讲的办法造出的边复盖 C' 实际上就是一个最小边复盖. 因此求最小边复盖的问题就完全解决了. 也就是说，如果要求一个图 G 的最小边复盖，可以先用上一章讲的办法求出 G 的一个最大匹配 M，然后按引理 11.1 的证明中讲的那样，对每一个未盖点，选一条与它关联的边，把这些边与 M 中的边合在一起，就得到 G 的一个最小边复盖了.

上面讲的求最小边复盖的方法，它的基本思想其实是很简单的. 我们希望选取尽量少的边把一个图的所有顶点都盖住，第一条边不管怎样选一定恰好盖住两个顶点，第二条边如果选得与第一条边没有公共端点，就又可以多盖住两个顶点，而如果与第一条边有公共端点，实际上只多盖住了一个顶点，……，一般说来，如果选取了 s 条边，这些边两两都没有公共端点，能盖住的顶点就最多(从而选取的边也就可能最少)，即盖住了 $2s$ 个. 因此，我们应该尽量多选一些两两没有公共

端点的边来复盖，很自然的应该先选取一个最大匹配 M，剩下的那些未盖点，想要用一条边盖住两个就不可能了，因此必须对一个顶点都用一条边来盖.

习　题

用本节讲的方法，分别给图 11.1 和 11.2 中的每个图各找一个最小边复盖.

结 束 语

这本小书就写到这里．当然，图论中的极值问题还有许多，这里就不再多讲了．举个例说，在这本书的引言中我们曾经介绍过一个"旅行售货员问题"，这个问题无疑是很有用的，但是我们却没有讲它的解法．为什么呢？或者说，作者是根据什么标准决定讲哪些，不讲哪些的呢？

标准是很清楚的．在第三章讲最短路问题时，我们曾经讨论过算法的有效性问题，在以后各章中，每讲完一个方法，我们也都简单地讨论一下这个方法是不是有效的．总起来说，这本书中讲的问题都是已经找到了有效解法的（并且解法还是较简单的）．而那些到目前为止还没有找到有效的解法的问题，就都没有在这本书中介绍．

"旅行售货员问题"就是这种没有找到有效解法的问题中最有名的一个，大约四十多年以前，就有人开始研究这个问题，并且已经找到了许多解这个问题的方法，但是所有这些解法需要的运算次数都不是图中顶点个数（即售货员需要去的城市的个数）n 的多项式，也就是说，这些解法都不是有效的．

有些问题，表面上看起来很简单，却也找不到有效的解法，例如"最大独立集问题"就是这样的一个问题．设 $G=[V, E]$ 是一个无向图，$V_1 \subseteq V$ 是 G 的若干个顶点的集合，如果属于 V_1 的任意两个顶点之间都没有边相连，我们就称

V_1 是一个独立集(独立集和匹配是有些相似的，匹配是若干条两两没有公共端点的边的集合，而独立集是两两没有边相连的顶点的集合). 很自然的可以提出一个极值问题，就是怎样把一个图 G 的包含顶点的个数最多的独立集找出来？这个问题与最大匹配问题很相似. 但是有趣的是，最大匹配问题有有效的解法，而最大独立集问题却没有找到有效的解法.

这类问题还可以举出很多，这里不多举了. 对于这类尚未找到有效解法的问题，当然还有许多研究工作可以做. 例如可以试试是否能找到有效的算法(当然这是很困难的)，或者证明一定不存在有效的算法(这也是很难的). 另外还可以研究近似解法，或者研究能不能不用第三章第 4 小节中提出的标准而用其他的标准来衡量解法的好坏等等. 当然，对于已找到有效算法的问题，也还有许多研究工作可以做.

总之，图论中的极值问题既有广泛的用途，而且对于有志于搞研究工作的青年人来说，还是一个大有用武之地的课题.